

X961115237

Reservado

UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR DE ECONOMIA E GESTÃO



MESTRADO EM: MATEMÁTICA APLICADA À ECONOMIA E À GESTÃO

**Minimização do Número de Más Viragens na Recolha
de Resíduos Sólidos Urbanos**

Carla Maria Santana Duarte Correia da Cunha

Orientação: Maria Cândida Vergueiro Monteiro Cidade Mourão

Júri:

Presidente: Doutora Margarida Maria Gonçalves Vaz Pato

Vogais: Doutor Miguel Fragoso Constantino

Doutora Maria Cândida Vergueiro Monteiro Cidade Mourão

Abril/2001

UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR DE ECONOMIA E GESTÃO



MESTRADO EM: MATEMÁTICA APLICADA À ECONOMIA E À GESTÃO

**Minimização do Número de Más Viragens na Recolha
de Resíduos Sólidos Urbanos**

Carla Maria Santana Duarte Correia da Cunha

Orientação: Maria Cândida Vergueiro Monteiro Cidade Mourão

Júri:

Presidente: Doutora Margarida Maria Gonçalves Vaz Pato

Vogais: Doutor Miguel Fragoso Constantino

Doutora Maria Cândida Vergueiro Monteiro Cidade Mourão

Abril/2001

Glossário

Abreviaturas

CARP	Problema de determinação de rotas óptimas para veículos de capacidade limitada, com procura nos arcos (<i>Capacitated Arc Routing Problem</i>);
CE	Circuito Euleriano;
CGRP	Problema de determinação de rotas óptimas para veículos de capacidade limitada (<i>Capacitated General Routing Problem</i>) com procura nos nodos e nos arcos;
CML	Câmara Municipal de Lisboa;
ETRS	Estação de Tratamento de Resíduos Sólidos;
PMV	Problema da minimização do número de Más Viragens;
PRRS	Problema da Recolha de Resíduos Sólidos urbanos;
VRP	Problema de determinação de rotas óptimas para veículos de capacidade limitada, com procura nos nodos (<i>Vehicle Routing Problem</i>).

Termos

Arcos, Circuito Euleriano, Grafo Euleriano, Frete, Má viragem, Nodos ou vértices, Rota de um veículo.

Simbologia

$ A $	Cardinal do conjunto A;
c.c.	caso contrário;
♦	Fim de exemplo;
❖	Fim de prova.

Resumo

O problema da determinação de percursos a efectuar por veículos afectos à remoção de resíduos sólidos urbanos pode ser visto como um problema de optimização de rotas, com procura nos arcos e restrições adicionais. Usualmente os algoritmos para determinar soluções admissíveis para este tipo de problemas não têm em conta o tipo de viragem que os veículos são obrigados a fazer. Dadas as características dos veículos envolvidos na recolha de resíduos, entende-se por *má viragem* uma inversão de marcha ou uma viragem à esquerda. Na prática é, em geral, exigida a *minimização do número de más viragens*, devendo mesmo algumas ser impossibilitadas, quer por se tornarem perigosas, quer por questões de regras de trânsito.

Neste trabalho o problema da minimização do número de más viragens é definido e modelizado como um problema de redes. São desenvolvidos métodos para a determinação de soluções admissíveis.

Os métodos implementados são testados num conjunto de problemas teste gerados aleatoriamente. Como se mostra, os resultados podem ser considerados bastante bons, dado permitirem obter soluções com um baixo número de más viragens sendo evitadas a quase totalidade das inversões de marcha proibidas. Muito embora a heurística se baseie na expansão da rede nos nodos onde as inversões de marcha não são permitidas, os resultados permitiram concluir que nem todos os nodos têm que ser expandidos para que as respectivas inversões de marcha sejam impossibilitadas.

Palavras Chave: Problemas de optimização de rotas; Circuito Euleriano; Heurísticas; Soluções admissíveis; Má viragem e inversão de marcha.

Abstract

Finding the optimal set of tours for vehicles dedicated to household refuse collection can be looked at as an arc routing problem with additional constraints. Typically, the algorithms used in determining feasible solutions to these problems do not account for the type of turn that the vehicles actually must perform. Given the specific characteristics of the vehicles assigned to household refuse collection, a *bad-turn* translates to a U-turn or to a left-turn. In practical terms it is generally mandatory to minimize the total number of *bad-turns*, some of which really must be avoided either due to its intrinsic danger or to traffic regulation restrictions.

Throughout this thesis the problem of minimizing the number of *bad-turns* is defined and modeled as a network problem. Some methods are developed to determine feasible solutions. These methods have been tested over a set of randomly generated problems.

It is shown that the overall results are to be considered pretty good since they represent feasible solutions with a very low number of *bad-turns*, where nearly every forbidden *U-turns* are avoided.

Key Words: Capacitated arc routing problems; Eulerian circuit; Heuristics; Feasible solutions; Bad and U-turn.

Índice

Cap.1	Introdução	9
Cap.2	Problema da Minimização do Número de Más Viragens	14
2.1	Introdução	14
2.2	Modelização em Redes e Notação	16
2.3	Enquadramento do Problema	35
Cap.3	Método Heurístico	38
3.1	Introdução	38
3.2	Método de Inspeção	40
3.2.1	Introdução	40
3.2.2	Construção da Matriz de Prioridades	41
3.2.3	Identificação de um Circuito Euleriano	44
3.3	Alteração do Circuito Euleriano	51
3.4	Ilustração	55
Cap.4	Resultados Computacionais	62
Cap.5	Conclusões	80
	Referências Bibliográficas	82

Lista de Figuras e Tabelas

Figura	Título	Página
1.1	Representação de uma rota admissível para um veículo.	12
2.1	Ilustração de más viragens.	18
2.2	Proibição de 1 inversão de marcha num nodo de grau 2.	22
2.3	Proibição de 1 inversão de marcha num nodo de grau 3.	23
2.4	Nodo ℓ de grau 2 e com $\text{inv}(\ell)=2$.	24
2.5	Ampliação da rede para a proibição de 2 inversões.	24
2.6	Nodo ℓ de grau 5, com $\text{inv}(\ell)=2$.	24
2.7	Rede expandida para a proibição de 2 inversões num nodo de grau 5.	25
2.8	Ampliação da rede para a proibição de 3 inversões.	26
2.9	Rede de estradas.	33
2.10	Grafo associado à rede de estradas.	33
3.1	Construção de Dir para o nodo i .	41
3.2	Grafo Euleriano, G .	44
3.3	Rede Euleriana	55
3.4	Nodo 7 expandido	60
Tabela		
3.1	Resumo das viragens efectuadas em CE_i	59
3.2	Resumo das viragens efectuadas em CEf .	61
4.1	Resultados do algoritmo <i>Inspecção</i> em redes de 17 nodos.	64
4.2	Viragens à esquerda em problemas com 17 nodos.	65
4.3	Inversões de marcha em problemas com 17 nodos.	66
4.4	Resultados do algoritmo <i>Inspecção</i> em redes de 26 nodos.	68
4.5	Viragens à esquerda em problemas com 26 nodos.	69
4.6	Inversões de marcha em problemas com 26 nodos.	70
4.7	Resultados do algoritmo <i>Inspecção</i> em redes de 37 nodos	71
4.8	Viragens à esquerda em problemas com 37 nodos.	72
4.9	Inversões de marcha em problemas com 37 nodos.	73
4.10	Resultados do algoritmo <i>Inspecção</i> em redes de 50 nodos.	74
4.11	Viragens à esquerda em problemas com 50 nodos.	75
4.12	Inversões de marcha em problemas com 50 nodos.	76
4.13	Resumo do número de más viragens	77
4.14	Resumo da alteração do número de inversões de marcha	78

Agradecimentos

À minha orientadora, Professora Doutora Maria Cândida Mourão, pela sua dedicação e disponibilidade postas na elaboração deste trabalho.

Aos meus pais, por todo o apoio que sempre me deram e por todos os sonhos que me permitiram ter.

Ao meu marido José Manuel, por tudo.

*“Felizes aqueles que se divertem com
problemas que educam a alma e elevam
o espírito.”*

Fénélon

Abril de 2001
Carla Correia Cunha

Capítulo 1. Introdução

Na optimização de rotas, em geral, considera-se como objectivo a minimização da distância (custo, tempo, etc.) total. Assim, pretende-se determinar um conjunto de percursos a efectuar por uma frota de veículos, de capacidades limitadas, que, minimizando a distância total percorrida, satisfaça um certo conjunto de restrições. As condições a considerar influenciam a modelização e, conseqüentemente, o tipo de abordagem. Entre as mais comuns, destacam-se as restrições de satisfação de ofertas e/ou procura dos clientes podendo estes, por sua vez, estar localizados num número discreto de pontos, tornando aconselhável o recurso a modelos nos vértices (*Vehicle Routing Problem–VRP*), ou localizados de uma forma contínua, indicando para modelos nos arcos (*Capacitated Arc Routing Problem–CARP*). Casos há em que são considerados os dois tipos de localizações dos clientes (discreta e contínua) recorrendo-se ao modelo geral (*Capacitated General Routing Problem–CGRP*). No livro editado por Ball et al. (1995) são abordados diversos problemas de determinação de rotas com procura nos arcos ou nos nodos. Em geral, são problemas NP–difíceis, ou seja, não se conhecem e pensa-se que não é possível encontrar algoritmos que forneçam a sua solução óptima em tempo polinomial.

O CARP é um problema de muita aplicabilidade prática, tendo sido foco de diversos estudos, como por exemplo os artigos de Marks e Liebman (1970); Beltrami e Bodin

(1974); Bodin e Kursh ((1978) e (1979)); Male e Liebman (1978); Stern e Dror (1979), mesmo antes de 1981, ano em que foi definido por Golden e Wong (1981). Já posteriormente continuou a despertar atenções, como, por exemplo, em Golden et al. ((1981) e (1983)); Chapleau et al. (1984); Roy e Rousseau (1988); Benavent et al. ((1990) e (1992)); Pearn (1991); Saruwatari et al. (1991); Álvarez-Valdés et al. (1993); Eglese e Li ((1992) e (1996)); Li e Eglese (1994); Greistorfer (1994a).

Como referido, em geral, a determinação de rotas para veículos é feita com o objectivo da minimização do custo total envolvido. Este trabalho surgiu na sequência de um problema de optimização de rotas para a recolha de resíduos sólidos (PRRS) na cidade de Lisboa.

Assim, considerando, por um lado, o objectivo da determinação de rotas com um custo total de recolha mínimo, pretende-se, por outro lado, minimizar certo tipo de viragens. Surgindo frequentemente dos modelos em que se minimiza apenas o custo, certo tipo de viragens dificultam a implementação prática das soluções associadas a esses modelos dadas as características dos veículos em causa.

O problema da minimização do número de más viragens, como por exemplo virar à esquerda ou a inversão de marcha, tem merecido atenção em alguns estudos. A título ilustrativo salientem-se os artigos de Bodin e Kursh ((1978) e (1979)), de McBride (1982)), de Roy e Rousseau (1988) e de Shuster e Schur (1974).

Genericamente, o problema base que deu origem a este trabalho, ou seja, a recolha de resíduos sólidos urbanos em Lisboa, pode ser resumido como se segue. A Câmara Municipal de Lisboa (CML) tem um certo número de veículos para a remoção de

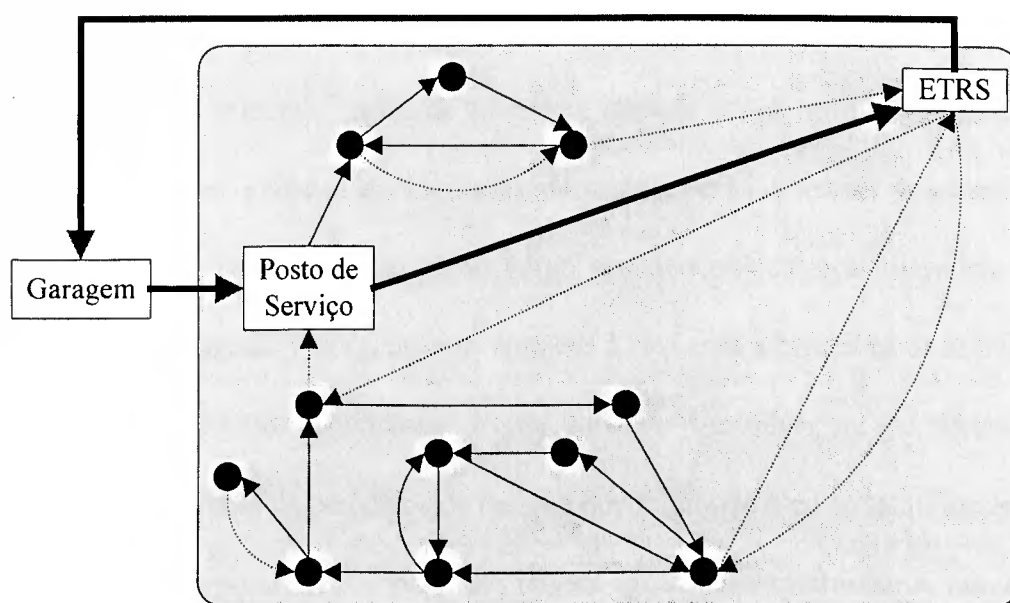
resíduos que são depositados em contentores de pequenas capacidades (120 ℓ ou 240 ℓ) localizados à porta de cada edifício da cidade. Neste caso, deve considerar-se um modelo com procura nos arcos, pois por rua existem vários contentores (no geral um ou dois por edifício).

Os veículos que efectuam a recolha de resíduos sólidos em Lisboa partem da garagem, na Av. 24 de Julho, levando apenas o motorista. A restante tripulação de cada veículo junta-se-lhe no Posto de Serviço da zona onde esse veículo vai fazer a recolha. O número de veículos afectos à recolha, em cada Posto de Serviço, é fixo antecipadamente.

Em determinados locais, previamente estabelecidos ou escolhidos pela tripulação, a recolha tem que ser interrompida, de forma a que não sejam ultrapassadas as capacidades dos veículos. Assim, e em geral uma a duas vezes por recolha, cada veículo dirige-se à Estação de Tratamento de Resíduos Sólidos (ETRS) para despejar os resíduos recolhidos.

Terminada a recolha, e antes do veículo despejar os resíduos na ETRS, pela última vez, a tripulação é deixada no Posto de Serviço respectivo. O motorista tem que finalmente levar o veículo de regresso à garagem. A figura 1.1, retirada de Mourão (1997), representa uma rota admissível para um veículo.

Figura 1.1: Representação de uma rota admissível para um veículo.



Note-se que as setas mais carregadas da figura envolvem apenas a determinação de caminhos mais curtos entre os respectivos pontos, pois não é efectuada recolha nestes percursos. Contudo, a parte sombreada corresponde a um problema de “difícil” resolução, dado ser necessário ter em conta tanto a capacidade do veículo como as quantidades de resíduos a recolher em cada rua, como afirmam Golden e Wong (1981). Assim, o problema relevante de recolha está representado apenas na parte sombreada da figura.

A optimização do percurso dos veículos tem como principal objectivo a minimização da distância total percorrida na recolha pela frota de veículos. A capacidade limitada dos veículos disponíveis origina a introdução de restrições de capacidade no problema cujas características específicas fazem com que seja possível a sua modelização se consideradas restrições adicionais num CARP (Mourão (1997)).

Contudo, em certos cruzamentos, existindo mais de uma alternativa para prosseguir a recolha, não se deve considerar indiferente a sequência de ruas escolhida. Como é óbvio, mesmo não existindo regras de trânsito a impedir certas sequências de ruas na recolha, deverão ser evitadas as inversões de marcha e as viragens à esquerda. O problema em foco nesta tese tem então como segundo objectivo a minimização do número de más viragens (PMV), ou seja viragens à esquerda e inversões de marcha, na identificação de percursos admissíveis. Assim, supondo disponível um mecanismo para determinar um conjunto de percursos de recolha que minimize o custo total, pretende-se neste trabalho incorporar ferramentas que tentem igualmente minimizar o número de más viragens. Como se verá, as inversões de marcha não permitidas pelas regras de trânsito serão mesmo impossibilitadas.

Inicia-se o segundo capítulo com uma modelização em redes do PMV. São também apresentadas algumas definições bem como a principal notação utilizada ao longo do texto. No final deste capítulo é feito o enquadramento do PMV, sendo descritas outras abordagens que constam da literatura disponível.

O método heurístico desenvolvido e implementado no âmbito desta tese constitui o terceiro capítulo. No quarto capítulo são apresentados e analisados os resultados obtidos para certos problemas teste. A terminar a tese são formuladas as principais conclusões sendo apontadas algumas sugestões para trabalho futuro.

Capítulo 2. Problema da Minimização do Número de “Más” Viragens

2.1. Introdução

O problema descrito no capítulo 1, PMV, em foco neste trabalho, tem como objectivo obter um conjunto de percursos admissíveis para os veículos que efectuem a recolha de resíduos sólidos urbanos que, minimizando o custo total, minimize ainda um certo tipo de viragens, denominadas por más viragens.

Uma forma de obter percursos admissíveis para os veículos consiste na partição de um percurso gigante em vários sub-percursos compatíveis com as capacidades dos veículos. Assim, se o percurso gigante for formado por um número pequeno de más viragens, o conjunto final de percursos a atribuir aos diferentes veículos será também formado por um baixo número de más viragens. Pois, sendo o percurso gigante subdividido com introdução de caminhos até à ETRS e estando esta afastada da zona de recolha, tais partições não provocam alteração ao número de más viragens.

Um percurso gigante pode ser facilmente identificado numa rede Euleriana, ou seja, uma rede em que todos os vértices têm grau interno igual ao seu grau externo (Ford e

Fulkerson, (1974)). Por sua vez, a criação de uma rede Euleriana relacionada com a rede de recolha de resíduos sólidos, para o problema em estudo nesta tese, foi construída por Mourão (1997). A identificação de um circuito Euleriano, ou seja, um circuito que passe uma e uma só vez por cada arco da rede, com um número “reduzido” de más viragens foi feita por alteração do algoritmo de Gondran e Minoux (1984) na escolha do próximo arco a considerar no circuito, como se verá no terceiro capítulo.

Contudo, certo tipo de viragens, por serem mais perigosas para o trânsito, são mesmo impossibilitadas. No próximo ponto explica-se a construção de uma rede, a partir da rede Euleriana que se supõe dada, onde se impossibilitam inversões de marcha. São ainda apresentadas algumas definições e a principal notação utilizada ao longo desta tese.

No último ponto deste capítulo será feito um breve enquadramento do PMV com trabalhos que têm surgido no tratamento deste tipo de problemas.

2.2. Modelização em Redes e Notação

Neste ponto pretende construir-se, a partir da rede Euleriana que se supõe dada, uma rede em que sejam eliminadas as inversões de marcha que, por algum motivo, são consideradas impossíveis.

A modelização será antecedida de alguns conceitos e da principal notação que se utilizará ao longo do presente texto.

Designa-se por *rota de um veículo* todo o seu percurso desde que sai do posto de Serviço até lá voltar para deixar a tripulação.

Uma rota de um veículo inclui, pelo menos, dois *fretes*: o primeiro, do Posto de Serviço à Estação de Tratamento de Resíduos Sólidos (ETRS), denomina-se por *frete inicial*; e o último, da ETRS ao Posto de Serviço é designado por *frete final* do veículo. Eventualmente, uma rota de um veículo inclui ainda *fretes intermédios*, da ETRS à ETRS.

O mapa da zona em estudo pode ser representado por um grafo orientado, $G = (V, A)$, com $n+1$ vértices ($|V| = n+1$), em que:

- O vértice 1 representa o Posto de Serviço;
- O vértice $n+1$ representa a ETRS;
- Os restantes vértices representam cruzamentos, entroncamentos ou extremos de ruas sem saída (“becos”);

- Cada arco $a = (i, j) \in A$, representa um segmento de rua. O sentido de cada rua é indicado pela orientação do arco, sendo uma rua com dois sentidos representada por dois arcos orientados: (i, j) e (j, i) .

Nas ilustrações desta tese, um arco de procura, ou seja, uma rua onde existem resíduos a recolher, é representado por um segmento de recta a cheio. Um segmento de recta a tracejado indica um arco que foi junto à rede para a tornar Euleriana, podendo portanto ser percorrido por um veículo em vazio, ou seja, sem efectuar recolha. O processo de obtenção de uma rede Euleriana, em que o grau interno de cada vértice seja igual ao seu grau externo, que minimize o custo total dos arcos adicionais foi o desenvolvido por Mourão (1997). O método baseia-se na resolução de um problema de transportes a partir da rede de procuras (rede em que se consideram apenas os arcos de procura) tomando como:

- origens, os vértices que tendo grau interno superior ao grau externo, só ficam equilibrados se forem vértices iniciais de arcos em vazio, com oferta representando a respectiva diferença entre graus;
- destinos, os vértices que tendo grau externo superior ao grau interno, só ficam equilibradas se forem vértices finais de arcos em vazio, com procura igual ao desequilíbrio respectivo;
- custos unitários de transporte de cada origem para cada destino, os custos dos caminhos de menor custo entre os vértices associados.

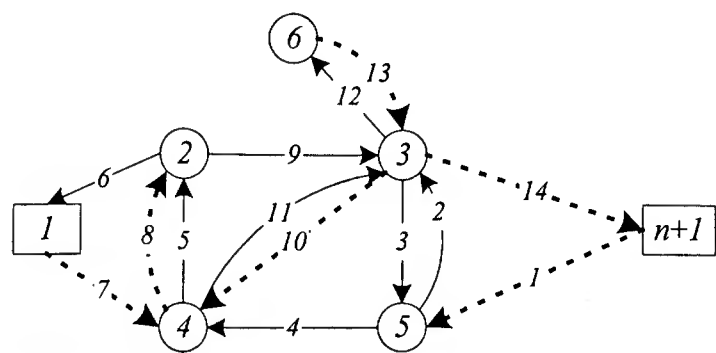
Cada unidade a transportar de uma origem para um destino, na solução óptima, representa um caminho mais curto que deve ser percorrido em vazio dessa origem

para esse destino. Tenta-se assim obter uma rede que, sendo equilibrada, tenha um baixo número de caminhos (de menor custo) a percorrer em vazio. Note-se que a rede resultante, sendo Euleriana, pode ter associado um multigrafo, ou seja, um grafo com dois ou mais arcos entre dois vértices.

Define-se **má viragem** num cruzamento, quando um veículo é obrigado a fazer uma inversão de marcha nesse cruzamento ou é obrigado a seguir para a rua que se encontra mais à esquerda. Este tipo de viragem considera-se “**má**” devido às características dos veículos envolvidos na recolha de resíduos que a torna em manobra perigosa. Uma viragem é tanto mais perigosa quanto mais à esquerda o veículo tiver que virar, podendo portanto existir diferentes tipos de más viragens. Sendo óbvio que uma inversão de marcha num nodo i corresponde a uma sequência de arcos de sentidos opostos $(i, j), (j, i)$, os restantes tipos de más viragens terão de ser definidos à partida para cada rede, pois o número de ruas concorrentes a cada cruzamento não determina o tipo de viragem.

Exemplo 2.1: Considere-se a seguinte sequência de arcos a ser percorrida por um veículo no seu frete inicial (arcos de 7 a 14) e no seu frete final (arcos de 1 a 6).

Fig. 2.1: Ilustração de más viragens.





Se o veículo efectuar o percurso seguindo a sequência de arcos 2, 3 está a efectuar uma má viragem, pois trata-se de uma inversão de marcha no vértice 3. Por outro lado, seguir do arco 5 para o arco 6 corresponde, neste caso, a uma viragem à esquerda, a evitar, tanto quanto possível. ♦

Para além de se considerar que todas as inversões representam uma má viragem, existem inversões e viragens não permitidas pelas regras de trânsito, que devem portanto ser impossibilitadas, tanto quanto possível.

Note-se que não admitir uma inversão de marcha não significa que não se possa ter, numa qualquer solução, os arcos representativos dessa inversão a serem percorridos pelo mesmo veículo, durante o mesmo frete, só não os poderá percorrer em sequência.

Exemplo 2.2: Retomando o exemplo anterior, se a sequência de arcos 2, 3 for proibida, ou seja, se for proibido inverter no nodo 3, não se pode ter esta sequência de arcos, mas pode ter-se, por exemplo, a sequência (2, 10, 5, 9, 3), ou seja os arcos 2 e 3 são percorridos, no mesmo frete de um veículo, mas não em sequência. ♦

Como referido, é possível a construção de um conjunto de fretes admissível a partir de um circuito Euleriano, CE. Influenciado pelo CE identificado, o conjunto de fretes criado, será tanto melhor quanto menos viragens más fizerem parte do circuito base. O problema em foco nesta tese é então o da identificação de um CE com o menor número possível de más viragens.

Por outro lado, num grafo Euleriano, existe sempre pelo menos um CE. Como anteriormente referido, nesta tese supõe-se dada uma rede Euleriana obtida com base no grafo de procuras (com as ruas em que existem resíduos a recolher) e no grafo inicial, representativo do mapa da zona de recolha.

Assim, pretende identificar-se um CE que evite as más viragens no grafo Euleriano

$G_E = (V, A_E)$, onde

$A_E = A_R \cup A_{PT}$, sendo A_R o conjunto de arcos de procura (onde existem resíduos a recolher) e A_{PT} o conjunto de arcos que podem ser percorridos em vazio (ou seja, sem ser efectuada recolha) que resulta, como se explicou, da resolução de um problema de transporte definido com o objectivo de equilibrar o grafo.

Como se verá, no próximo capítulo, as viragens à esquerda são minimizadas na própria identificação do CE, impondo regras específicas na escolha do próximo arco a considerar para ser incluído no circuito em construção. Como se explica de seguida, a eliminação de inversões de marcha nas soluções de problemas de determinação de rotas, pode ser conseguida com a transformação do grafo Euleriano, sobre o qual se vão construir as rotas, num grafo que impossibilite tais sequências de arcos na identificação de circuitos Eulerianos.

No que se segue representa-se por:

$G_E = (V, A_E)$ o grafo Euleriano, com $|V| = n + 1$ e $|A_E| = m$.

$g^-(v)$ ($g^+(v)$) o grau interno (externo) do vértice $v \in V$ em G_E , ou seja, o número de arcos com vértice final (inicial) em v .

$g(v) = g^-(v) = g^+(v)$ o grau do vértice $v \in V$ em G_E . Sendo G_E um grafo Euleriano, por simplificação, falar-se-á só em grau do vértice, pois todos os vértices estão equilibrados.

$cruza = \underset{v \in V}{Máx} \{2g(v)\}$ o número máximo de viragens em G_E .

$inv(v)$ o número de inversões de marcha a impossibilitar em $v \in V$. Note-se que $inv(v) \leq g(v)$.

As regras de trânsito impõem não só proibições de inversão de marcha, como limitações a certas manobras, como, por exemplo, viragens à esquerda e à direita. Contudo, nesta tese, tenta-se apenas minimizar as viragens à esquerda e impedir certas inversões de marcha, ficando as restantes proibições para possíveis investigações futuras.

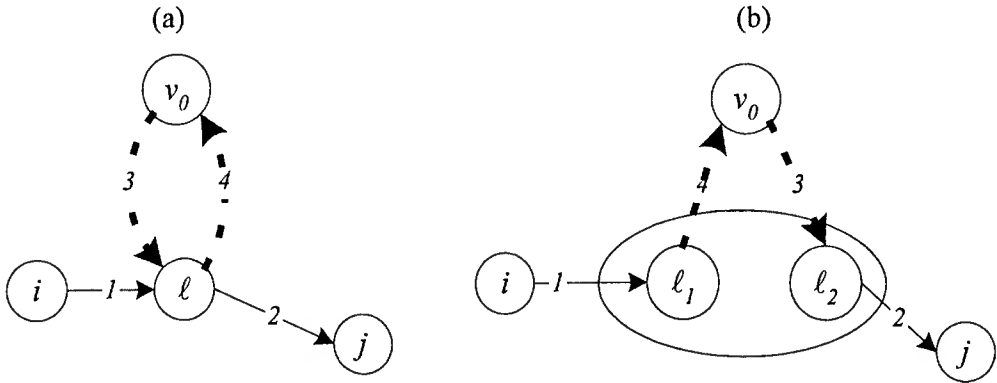
Para impossibilitar as inversões de marcha em certos cruzamentos obtém-se um grafo expandido em que o cruzamento em questão é ampliado, tendo em conta o número de inversões não possíveis. No exemplo seguinte ilustram-se algumas dessas situações.

Exemplo 2.3: Represente-se por $\ell \in V$ um cruzamento em que não se admitem as inversões de marcha associadas às sequências de arcos $((v_k, \ell), (\ell, v_k))$.

A) Se $k = 0$, pretende-se impedir a sequência $((v_0, \ell), (\ell, v_0))$, sendo $inv(\ell) = 1$.

(i) Seja $g(\ell) = 2$ ⁽¹⁾, como na figura seguinte:

Fig. 2.2: Proibição de 1 inversão de marcha num nodo de grau 2.

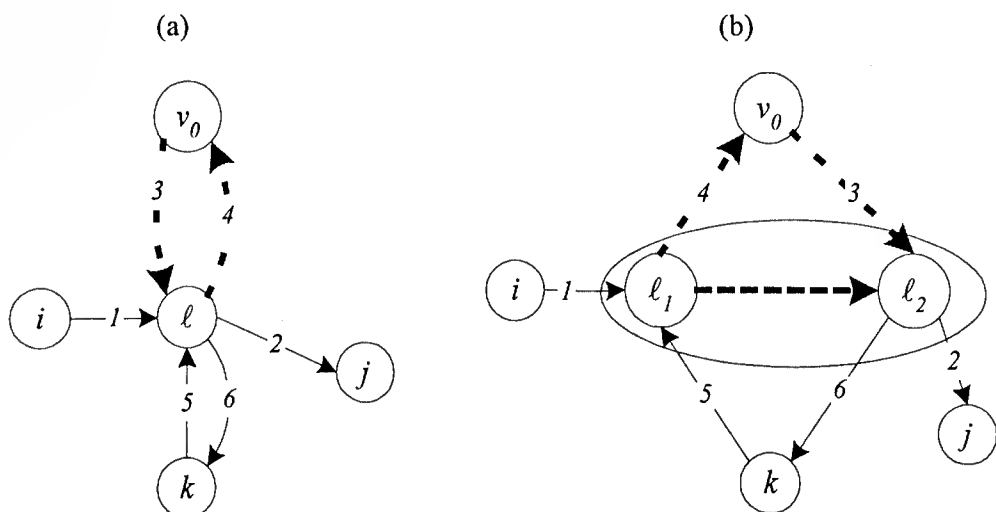


Por forma a impedir a sequência de arcos 3, 4 da figura 2.2(a), expande-se o vértice ℓ , em ℓ_1 e ℓ_2 , como na figura 2.2(b). Na rede expandida só se permitem as ligações correspondentes a $((v_0, \ell), (\ell, j))$ e $((i, \ell), (\ell, v_0))$. Propositadamente, para além de impedir a inversão de marcha impossibilita-se também a sequência $((i, \ell), (\ell, j))$, pois as duas estão relacionadas. Note-se que, tendo ℓ grau 2, ao arco 3 só pode seguir o arco 2 (e 4 só pode ser precedido de 1).

⁽¹⁾ Note-se que se $g(\ell) = 1 = inv(\ell)$ não é possível impedir a inversão. Logo, $g(\ell) - inv(\ell) \geq 1$.

(ii) Seja $g(\ell) = 3$ ($g(\ell) - \text{inv}(\ell) = 2$), como na figura seguinte:

Fig. 2.3: Proibição de 1 inversão de marcha num nodo de grau 3.

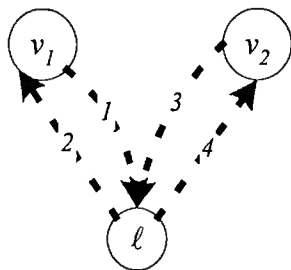


Embora semelhante ao caso anterior, como o grau de ℓ é 3, ao arco 3 pode suceder o arco 2 ou o 6, mas nunca o arco 4. Do mesmo modo, o arco 4 só pode ser antecedido pelo arco 1 ou pelo arco 5. É então possível obter caminhos que, passando por ℓ , tenham início e fim num vértice diferente de v_0 , não se podendo por isso eliminar as sequências como em (i). Tal como na figura 2.3(b), o grafo expandido deve então incluir o arco (ℓ_1, ℓ_2) . Assim, quando $g(\ell) > 2$ consideram-se $g(\ell) - 2$ cópias do arco (ℓ_1, ℓ_2) no grafo expandido, mantendo assim o equilíbrio destes vértices.

B) Se $k = 1, 2$, pretende-se impedir as duas sequências $((v_1, \ell), (\ell, v_1))$ e $((v_2, \ell), (\ell, v_2))$, sendo $\text{inv}(\ell) = 2$.

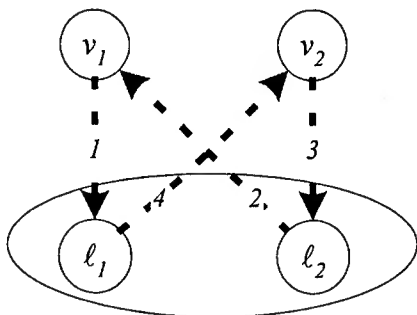
(i) Seja $g(\ell) = inv(\ell)$, como na figura seguinte:

Fig. 2.4: Nodo ℓ de grau 2 e com $inv(\ell) = 2$.



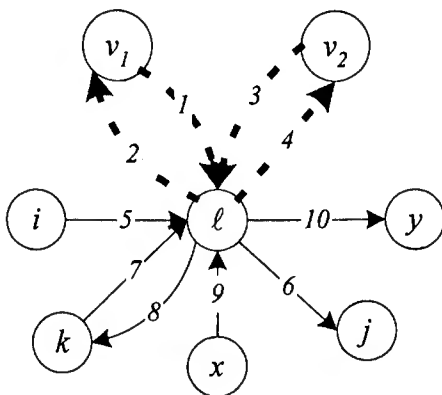
A expansão do vértice ℓ é feita de forma a evitar as duas inversões de marcha, mantendo todos os restantes caminhos inicialmente possíveis. Assim, criam-se dois nodos ℓ_1 e ℓ_2 , para manter as ligações aos nodos v_1 e v_2 , impedindo as inversões de marcha, resultando:

Fig. 2.5: Ampliação da rede para a proibição de 2 inversões.



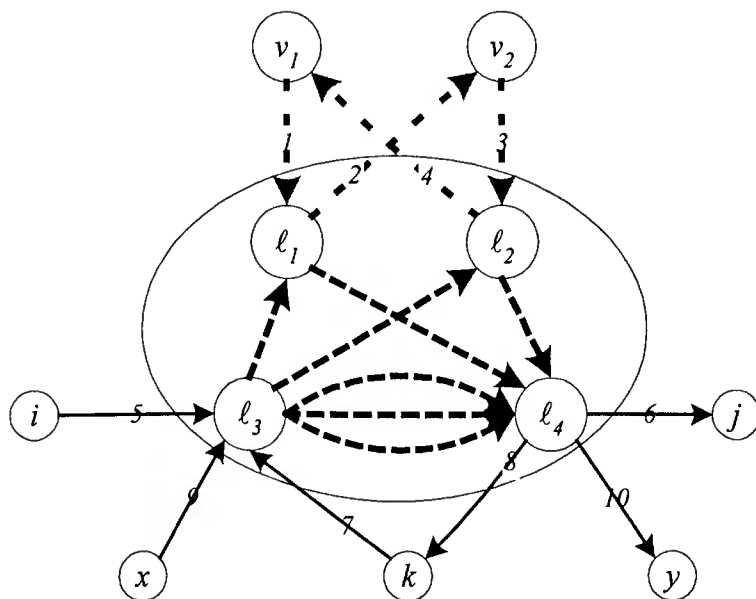
(ii) Seja $g(\ell) = 5 > inv(\ell) = 2$, como na figura seguinte:

Fig. 2.6: Nodo ℓ de grau 5, com $inv(\ell) = 2$.



Além de considerar as ligações do caso anterior, há agora que ter em atenção os restantes vértices (i, j, k, x e y). Criam-se, então, mais duas réplicas de ℓ , ℓ_3 e ℓ_4 . O nodo ℓ_3 é utilizado para manter os arcos com vértice final em ℓ e ℓ_4 para os que têm ℓ como nodo inicial, resultando a rede ampliada seguinte.

Fig. 2.7: Rede expandida para a proibição de 2 inversões num nodo de grau 5.



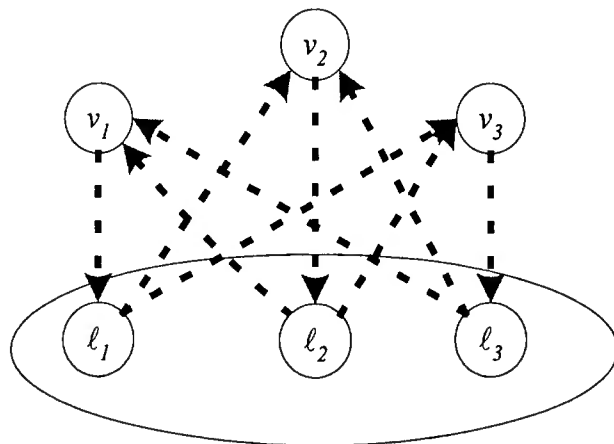
Podendo, em certas situações, ser evitado, o arco (ℓ_3, ℓ_4) considera-se sempre por permitir uma implementação computacional mais eficiente. Por outro lado, e tendo em conta que ao construir o circuito Euleriano os arcos já utilizados vão sendo retirados do conjunto de arcos a incluir em CE, são sempre feitas $g(\ell) - inv(\ell)$ réplicas de (ℓ_3, ℓ_4) . Permite-se assim manter os caminhos iniciais que passam por ℓ ao identificar o CE. Note-se que o desequilíbrio dos nodos ℓ_3 e ℓ_4 não afecta a identificação de CE, pois após serem identificados $g(\ell)$ caminhos que passam por ℓ já não se consideram os arcos, entre nodos réplicas de ℓ , por utilizar.

C) Seja $k = 1, 2, 3$. Há que impedir as três inversões: $((v_k, \ell), (\ell, v_k))$, sendo

$inv(\ell) = 3$. Se $g(\ell) = 3$ procede-se analogamente ao caso da figura 2.5, criando

3 réplicas de ℓ . Resulta então a rede da figura seguinte.

Fig. 2.8: Ampliação da rede para a proibição de 3 inversões.



A generalização, aos casos em que $g(\ell) > 3$, é também semelhante à anterior, sendo

sempre necessário mais duas réplicas de ℓ , quando $g(\ell) > inv(\ell)$. ♦

Assim, e em resumo, se existe apenas uma inversão a impedir em ℓ , $inv(\ell) = 1$, expande-se o nodo ℓ em dois (ℓ_1 e ℓ_2) e fazem-se as ligações entre os nodos por forma a evitar a inversão de marcha proibida. Se $g(\ell) > 2$, fazem-se $g(\ell) - 2$ réplicas do arco (ℓ_1, ℓ_2) . Se existem 2 ou mais inversões a impedir em ℓ ($inv(\ell) > 1$), consideram-se $inv(\ell) + 2$ réplicas de ℓ , ℓ_i . Como se ilustrou no exemplo, os nodos réplica $\ell_1, \ell_2, \dots, \ell_{inv(\ell)}$, são utilizados para manter as ligações entre os nodos em que se pretendem proibir inversões, sem permitir as inversões não possíveis. Os dois vértices $\ell_{inv(\ell)+1}$ e $\ell_{inv(\ell)+2}$ são criados para manter todos os caminhos iniciais que passam por ℓ .

A expansão da rede Euleriana foi então feita com base no algoritmo que se descreve de seguida, onde, tal como no exemplo, se diferencia o caso de só ser impedida uma inversão de marcha no vértice ℓ , dos casos em que são proibidas duas ou mais inversões.

Algoritmo: *Expande Rede*

Passo 0. Seja J o conjunto de vértices onde existem limitações às inversões de marcha e $G_E = (V, A_E)$ o multigrafo Euleriano de partida;

Fazer $\bar{V} = V$, $\bar{A} = A_E$, $\bar{G} = (\bar{V}, \bar{A})$

Construir um vector com todas as sequências de arcos que representam inversões de marcha proibidas

Passo 1. Se $J = \emptyset$ **FIM.**

Passo 2. Seleccionar $\ell \in J$

Calcular $L = \begin{cases} inv(\ell) & se \quad inv(\ell) > 1 \\ 0 & se \quad inv(\ell) = 1 \end{cases}$

Represente-se por v_s os vértices ligados a ℓ por arcos onde não é permitida a

inversão de marcha, com $v_s = \begin{cases} v_1, \dots, v_L & se \quad L > 0 \\ v_0 & se \quad L = 0 \end{cases}$

Se $L = 0$ **fazer** $\bar{V} \leftarrow \bar{V} \setminus \{\ell\} \cup \{\ell_1, \ell_2\}$

c.c., $\bar{V} \leftarrow \bar{V} \setminus \{\ell\} \cup \{\ell_1, \dots, \ell_L, \ell_{L+1}, \ell_{L+2}\}.$

Passo 3. Substituir os arcos de \bar{A} incidentes em ℓ , excluindo os incidentes nos nodos v_s , ou seja,

$$\forall u \neq v_s : \exists (u, \ell) \in \bar{A}, \quad \bar{A} \leftarrow \bar{A} \setminus \{(u, \ell)\} \cup \{(u, \ell_{L+1})\}$$

$$\forall u \neq v_s : \exists (\ell, u) \in \bar{A}, \quad \bar{A} \leftarrow \bar{A} \setminus \{(\ell, u)\} \cup \{(\ell_{L+2}, u)\}$$

Passo 4. {Ligações aos vértices v_s }

$$\text{Se } L=0 \quad \text{fazer} \quad \bar{A} \leftarrow \bar{A} \setminus \{(v_0, \ell), (\ell, v_0)\} \cup \{(v_0, \ell_2), (\ell_1, v_0)\}$$

$$\text{Se } g(\ell) > 2 \text{ juntar a } \bar{A} \text{ } (g(\ell) - 2) \text{ cópias de } (\ell_1, \ell_2)$$

c.c., fazer:

$$\bar{A} \leftarrow \left(\bar{A} \setminus \bigcup_{k=1}^L \{(v_k, \ell)\} \right) \cup \left(\bigcup_{k=1}^L \{(v_k, \ell_k)\} \right)$$

$$\bar{A} \leftarrow \left(\bar{A} \setminus \bigcup_{k=1}^L \{(\ell, v_k)\} \right) \cup \left(\bigcup_{\substack{k=1 \\ k \neq j}}^L \bigcup_{j=1}^L \{(\ell_k, v_j)\} \right)$$

$$\bar{A} \leftarrow \bar{A} \cup \left(\bigcup_{k=1}^L \{(\ell_{L+1}, \ell_k), (\ell_k, \ell_{L+2})\} \right)$$

$$\text{Juntar a } \bar{A} \text{ } (g(\ell) - \text{inv}(\ell)) \text{ cópias de } (\ell_{L+1}, \ell_{L+2}).$$

Passo 5. $J \leftarrow J \setminus \{\ell\}$.

Passo 6. Voltar a 1.

Como se mostra de seguida, na rede assim obtida mantém-se todos os caminhos, que se pretendem manter, da rede inicial, impossibilitando as inversões não permitidas. Com o objectivo de simplificar e tendo em conta que a expansão do grafo é feita vértice a vértice, as provas são feitas supondo que $|J| = 1$, ou seja, que só existe um cruzamento

com limitações a inversões de marcha. A extensão para qualquer que seja a cardinalidade de J é possível.

Propriedade 1: Seja $\bar{G} = (\bar{V}, \bar{A})$ o grafo resultante de $G = (V, A)$, por aplicação do algoritmo *Expande Rede*, e $\ell \in V : g(\ell) = \text{inv}(\ell) \geq 2$ um vértice expandido. Considere-se que $|J| = 1$.

$\forall u, v \in V, u \neq v : (u, \ell), (\ell, v) \in A \Rightarrow$ Existe caminho de u para v em \bar{A}
que inclui apenas vértices réplicas de ℓ , u e v .

Prova: $g(\ell) = \text{inv}(\ell) \geq 2 \Rightarrow$ $L = \text{inv}(\ell)$ e $\bar{V} = V \setminus \{\ell\} \cup \{\ell_1, \dots, \ell_L, \ell_{L+1}, \ell_{L+2}\}$
passo 2

Tome-se quaisquer $(u, \ell), (\ell, v) \in A$ e considere-se, tendo em conta a notação do algoritmo, $v_i = u, v_j = v$, $i, j = 1, \dots, L, i \neq j$

$$\left. \begin{array}{l} u = v_i \Rightarrow (u, \ell_i) \in \bar{A} \\ \text{passo 4} \\ v = v_j \Rightarrow \exists_{i \neq j} (\ell_i, v) \in \bar{A} \\ \text{passo 4} \end{array} \right\} \Rightarrow (u, \ell_i), (\ell_i, v) \in \bar{A}$$

$\therefore \exists$ caminho de u para v em \bar{A} formado só com réplicas de ℓ . ❖

Propriedade 2: Seja $\bar{G} = (\bar{V}, \bar{A})$ o grafo resultante de $G = (V, A)$, por aplicação do algoritmo *Expande Rede*, e $\ell \in V : g(\ell) \geq 3$ um vértice expandido. Considere-se que $|J| = 1$.

$\forall u, v \in V, u \neq v : (u, \ell), (\ell, v) \in A \Rightarrow$ Existe caminho de u para v em \bar{A}
que inclui apenas vértices réplicas de ℓ , u e v .

Prova: Seja $\ell \in V : \text{inv}(\ell) \geq 1$

$$\text{A) Se } \text{inv}(\ell) = 1 \Rightarrow_{\text{passo 2}} L = 0, \quad \bar{V} = V \setminus \{\ell\} \cup \{\ell_1, \ell_2\}$$

$$(i) \quad u, v \neq v_0$$

$$\text{passo 3: } \left\{ \begin{array}{l} (u, \ell) \in A \Rightarrow (u, \ell_1) \in \bar{A} \\ (\ell, v) \in A \Rightarrow (\ell_2, v) \in \bar{A} \end{array} \right\} \Rightarrow (u, \ell_1), (\ell_1, \ell_2), (\ell_2, v) \in \bar{A}$$

$$g(\ell) \geq 3 \Rightarrow_{\text{passo 4}} (\ell_1, \ell_2) \in \bar{A}$$

$$(ii) \quad u = v_0, \quad v \neq v_0$$

$$\left\{ \begin{array}{l} u = v_0 \Rightarrow_{\text{passo 4}} (u, \ell_2), (\ell_1, u) \in \bar{A} \\ (\ell, v) \in A \Rightarrow_{\text{passo 3}} (\ell_2, v) \in \bar{A} \end{array} \right\} \Rightarrow (u, \ell_2), (\ell_2, v) \in \bar{A}$$

$$(iii) \quad u \neq v_0, \quad v = v_0$$

$$\left\{ \begin{array}{l} u \neq v_0 \Rightarrow_{\text{passo 3}} (u, \ell_1) \in \bar{A} \\ v = v_0 \Rightarrow_{\text{passo 4}} (v, \ell_2), (\ell_1, v) \in \bar{A} \end{array} \right\} \Rightarrow (u, \ell_1), (\ell_1, v) \in \bar{A}$$

$\therefore \exists$ caminho de u para v em \bar{A} formado só com réplicas de ℓ .

$$\text{B) Se } \text{inv}(\ell) > 1 \Rightarrow_{\text{passo 2}} L = \text{inv}(\ell), \quad \bar{V} = V \setminus \{\ell\} \cup \{\ell_1, \dots, \ell_L, \ell_{L+1}, \ell_{L+2}\}$$

Tendo em conta a propriedade anterior, falta provar a existência de caminhos apenas quando $g(\ell) > \text{inv}(\ell) = L \geq 2$.

Pelo passo 4, $\forall_{s=1, \dots, L}, (\ell_{L+1}, \ell_s), (\ell_s, \ell_{L+2}) \in \bar{A}$ e $(\ell_{L+1}, \ell_{L+2}) \in \bar{A}$.

$$(i) \quad u \neq v_s, v \neq v_s, \quad \forall s = 1, \dots, L$$

$$\left. \begin{array}{l} (u, \ell) \in A \xRightarrow{\text{passo 3}} (u, \ell_{L+1}) \in \bar{A} \\ (\ell, v) \in A \xRightarrow{\text{passo 3}} (\ell_{L+2}, v) \in \bar{A} \end{array} \right\} \Rightarrow (u, \ell_{L+1}), (\ell_{L+1}, \ell_{L+2}), (\ell_{L+2}, v) \in \bar{A}$$

$$(ii) \quad u = v_s, v \neq v_s, \quad \forall s = 1, \dots, L$$

$$\left. \begin{array}{l} u = v_s \xRightarrow{\text{passo 4}} (u, \ell_s) \in \bar{A} \\ (\ell, v) \in A \xRightarrow{\text{passo 3}} (\ell_{L+2}, v) \in \bar{A} \end{array} \right\} \Rightarrow (u, \ell_s), (\ell_s, \ell_{L+2}), (\ell_{L+2}, v) \in \bar{A}.$$

$$(iii) \quad u \neq v_s, v = v_s, \quad \forall s = 1, \dots, L$$

$$\left. \begin{array}{l} (u, \ell) \in A \xRightarrow{\text{passo 3}} (u, \ell_{L+1}) \in \bar{A} \\ v = v_s \xRightarrow{\text{passo 4}} \exists k \neq s (\ell_k, v) \in \bar{A} \wedge (\ell_{L+1}, \ell_k) \in \bar{A} \end{array} \right\} \Rightarrow (u, \ell_{L+1}), (\ell_{L+1}, \ell_k), (\ell_k, v) \in \bar{A}$$

$$(iv) \quad u = v_s \neq v = v_k, \quad \forall s, k = 1, \dots, L, s \neq k$$

$$\left. \begin{array}{l} u = v_s \xRightarrow{\text{passo 4}} (u, \ell_s) \in \bar{A} \\ v = v_k \xRightarrow{\text{passo 4}} (s \neq k \Rightarrow (\ell_s, v) \in \bar{A}) \end{array} \right\} \Rightarrow (u, \ell_s), (\ell_s, v) \in \bar{A}$$

$\therefore \exists$ caminho de u para v em \bar{A} formado só com réplicas de ℓ . ❖

Como justificado anteriormente, quando $inv(\ell) = 1$ e $g(\ell) = 2$, não se inclui o arco (ℓ_1, ℓ_2) , pois impedir a inversão de marcha implica, neste caso, impedir os caminhos via ℓ que poderiam utilizar este arco. Ou seja, nem todos os caminhos iniciais têm correspondência na rede expandida, dado o seu relacionamento com a inversão a impedir.

Na propriedade seguinte prova-se então que as inversões que se pretende são mesmo impedidas em \bar{G} .

Propriedade 3: Seja: $\bar{G} = (\bar{V}, \bar{A})$ o grafo resultante de $G = (V, A)$, por aplicação do algoritmo *Expande Rede*;

$\ell \in V$ um vértice expandido no algoritmo *Expande Rede*;

$v \in V : (v, \ell, v)$ um circuito não possível em G , correspondente a uma inversão de marcha proibida.

Então, não existe circuito que inclua v em \bar{G} incluindo apenas vértices réplicas do vértice ℓ e v .

Prova : $\ell \in V$

A) Se $inv(\ell) = 1 \Rightarrow v = v_0, \bar{V} = V \setminus \{\ell\} \cup \{\ell_1, \ell_2\}$.

As únicas ligações de v com as cópias de ℓ , são: (v, ℓ_2) e (ℓ_1, v) .

Como $(\ell_2, \ell_1) \notin \bar{A}$, então não existe em \bar{G} circuito contento v e apenas os vértices ℓ_2 e/ou ℓ_1 .

B) $inv(\ell) > 1$.

Seja $v = v_1, L = inv(\ell), \bar{V} = V \setminus \{\ell\} \cup \{\ell_1, \dots, \ell_L, \ell_{L+1}, \ell_{L+2}\}$.

Pelo passo 4, o único arco de \bar{A} com extremo inicial v e extremo final numa réplica de ℓ é $(v, \ell_1) \in \bar{A}$.

Por outro lado, $(\ell_1, v) \notin \bar{A}$, mas $(\ell_1, \ell_{L+2}) \in \bar{A}$ e é o único arco com vértice inicial ℓ_1 e extremo final num nodo réplica de ℓ . Então, existe caminho de v para ℓ_{L+2} em \bar{A} .

Contudo, pelo passo 3, ℓ_{L+2} só é extremo inicial de arcos que não têm nenhum nodo v_s (nem v , em particular) como extremo final, nem réplicas de ℓ , ou seja, $(\ell_{L+2}, v) \notin \bar{A}$ e $\forall_{i=1, \dots, L+1}, (\ell_{L+2}, \ell_i) \notin \bar{A}$

Então, não existe em \bar{G} circuito contendo v e apenas vértices réplicas do vértice ℓ . ❖

Neste trabalho, como se tem visto, é considerada uma inversão de marcha num nodo i quando um veículo vai de um nodo j para um nodo i e regressa imediatamente a j , ou seja, quando realiza uma sequência de arcos $(i, j), (j, i)$. No entanto, poderão existir inversões de marcha que não correspondam a arcos opostos como se ilustra de seguida.

Fig. 2.9: Rede de estradas.

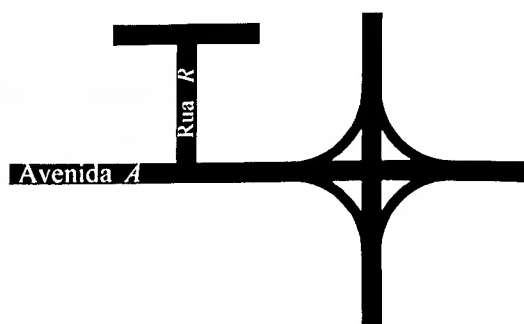
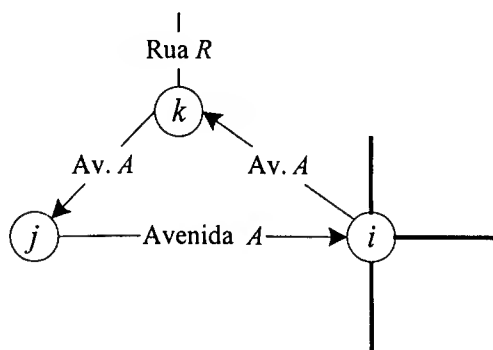


Fig. 2.10: Grafo associado à rede de estradas.



Os cruzamentos i e j estão ligados por uma avenida (Av. A) com separador central e, no sentido representado por (i, j) , existe um entroncamento (nodo k) a meio da avenida. Neste caso, o arco (i, j) é subdividido em dois, (i, k) e (k, j) , como na figura 2.10. Assim, a sequência de arcos $(j, i), (i, k)$ representa uma inversão de marcha em i e não é uma sequência de arcos opostos. Neste trabalho não são considerados este tipo de inversões de marcha, pois torna-se difícil identificá-las apenas através do grafo, não sendo portanto impedidas. No entanto, as duas sequências de arcos que na realidade

podem representar inversões de marcha são, como se verá, consideradas as viragens mais à esquerda, sendo portanto evitadas.

Em suma, o problema a resolver coincide com a identificação de um CE que minimize o número de viragens à esquerda na rede expandida, \overline{G} . Contudo, caso o número de cruzamentos com proibições de inversões seja elevado a rede \overline{G} pode ter dimensões muito maiores que a rede Euleriana inicial, G_E .

Como se verá no próximo capítulo, para evitar que tal aconteça, na heurística desenvolvida, optou-se por determinar primeiro o CE que minimize o número de viragens à esquerda em G_E e só expandir a rede nos vértices em que sejam feitas inversões não permitidas no CE encontrado. O contra desta abordagem é que a identificação de um CE numa rede em que são expandidos apenas alguns vértices pode originar novas inversões de marcha não permitidas. Tal faz com que, no pior caso, tenha que se expandir a rede $|J|$ vezes e, consequentemente, identificar $(|J| + 1)$ CE.

2.3. *Enquadramento do Problema*

Como anteriormente referido, o presente trabalho foca um sistema de recolha de resíduos sólidos urbanos. Estes sistemas são muito complexos, envolvendo abordagens a vários e diferentes níveis.

Marks e Liebman (1970) e, mais recentemente, Bodin et al. (1989) identificaram vários problemas relacionados com sistemas de recolha de resíduos sólidos urbanos, apresentando, já nessa altura, muitas referências bibliográficas. Entre os problemas mais relevantes destacam-se: *número e tipo de veículos* da frota (homogénea ou não); *número de trabalhadores* por veículo, tendo em conta épocas ou dias de semana com procuras muito distintas (por exemplo, Natal); *frequência de recolha*; *afecção de cada zona de recolha a frequências de recolha*; *identificação das melhores rotas de recolha*; *afecção dos veículos às zonas de recolha*; *localização e número das estações de tratamento*; *afecção das diferentes zonas de recolha a uma estação de tratamento*; *minimização dos "cruzamentos" de veículos*; *equilíbrio entre diferentes rotas*; *controle da geração de resíduos*; *eliminação dos resíduos*; etc.

Caruso et al. (1993) estudaram a parte do sistema de recolha posterior à recolha nos quarteirões considerando múltiplos objectivos, pois tentaram abordar, em simultâneo, localizações para estações intermédias e de estações de tratamento finais, o transporte da zona de recolha para as referidas estações intermédias (quando existem) e/ou finais, bem como tecnologias associadas (incineradoras; reciclagem; etc.).

Por sua vez, o problema de optimização de rotas de recolha, um dos mais focados ao longo dos tempos, pode ser encarado com diferentes abordagens dependendo das hipóteses iniciais: *procura aleatória* ou *determinística*; *procura uniforme* (limpeza de neve nas ruas) ou *não uniforme* (recolha de resíduos); *procura nos nodos* (Marks e Liebman (1970), Turner e Hougland (1975)) ou *nos arcos* (Álvarez-Valdés et al. (1993), Shuster e Schur (1974)); *rede não orientada* (ex., Beltrami e Bodin (1974), Bodin et al. (1989), Male e Liebman (1978)), *orientada* (Mourão (1997)) ou *mista* (Minieka (1979), Greistorfer (1994b)); tipo de *restrições de capacidade*; identificação dos *custos* que podem ou não ser considerados *variáveis*.

Não sendo obviamente possível tratar simultaneamente todos estes problemas é vulgar que estudos diferentes tenham como objecto apenas alguns dos aspectos identificados.

A recolha de resíduos sólidos urbanos foi já tema de alguns artigos. Porém, cidades diferentes originam a formulação de hipóteses também diferentes e, conseqüentemente, as abordagens podem considerar-se como únicas. A título ilustrativo salientem-se os trabalhos desenvolvidos para as cidades de: New York, por Beltrami e Bodin (1974); Oyster Bay, por Bodin et al. (1989); Cleveland, por Clark e Gillean (1975); Knoxville, por Male e Liebman (1978); Baltimore, por Marks e Liebman (1970); Izmir, por Or (1990). Mais recentemente, Álvarez-Valdés et al. (1993) apresentaram um sistema para recolha de resíduos que, segundo afirmam, se aplica a áreas residenciais afastadas do centro das cidades.

Na implementação prática dos sistemas de recolha de resíduos, o problema da minimização do número de más viragens, como por exemplo virar à esquerda ou a inversão de marcha, pode ser considerado como um dos mais importantes, sendo

também de difícil resolução. Entre os autores que focam a importância deste problema salientem-se, na década de 70: Shuster e Schur (1974), Bodin e Kursh ((1978) e (1979)), e na década de 80: McBride (1982), Roy e Rousseau (1988).

McBride (1982), para evitar certo tipo de viragens começa por resolver um problema de afectação em que os custos representam penalidades associadas às más viragens. Contudo esta abordagem levanta o problema da obrigatoriedade de incluir certo tipo de sequências nas rotas finais. Tendo em conta que os veículos têm capacidades limitadas, agrupar sequências de arcos desta forma pode tornar-se prejudicial, nomeadamente pode dificultar a obtenção de um conjunto de rotas equilibradas, em termos do peso total de cada rota, para os diferentes veículos.

Nesta tese optou-se por uma abordagem diferente para este problema. Assim, tenta-se impedir as más viragens, por regras específicas na identificação dos circuitos e por ampliação da rede inicial, como se explica no capítulo seguinte. Em aberto fica a comparação dos resultados obtidos neste trabalho com os que se obteriam com o método de McBride (1982).

Capítulo 3. Método Heurístico

3.1. Introdução

O objectivo deste capítulo é o da identificação de um circuito Euleriano, CE, na rede Euleriana G_E , que minimize o número de más viragens. Basicamente, o método proposto e implementado que se descreve de seguida – *Identifica CE* – pode considerar-se composto por três etapas: *Inspecção; Expande Rede; Altera CE*.

Com o método de inspecção pretende-se que a identificação do CE conduza à obtenção de circuitos em que o número de más viragens em cada vértice seja mínimo. São definidos diferentes tipos de viragens e tem-se sempre, como objectivo, virar o menos à esquerda possível.

Contudo, se no CE identificado, permanecerem inversões de marcha não admissíveis, num dado vértice, recorre-se ao método da rede expandida, em que é feita a expansão do nodo em causa. O CE é então alterado, tendo em conta o CE anterior e a rede expandida, em que não são admitidas as inversões de marcha.

A identificação do CE que minimiza o número de más viragens foi então feita com base no algoritmo que se descreve de seguida.

Algoritmo: *Identifica CE*

Passo 0: Seja G_E um grafo Euleriano.

Passo 1: Identificar em G_E um CE (algoritmo *Inspecção*, ponto 3.2.3) que tenha em atenção as más viragens.

Passo 2: Seja J o conjunto de nodos com inversões de marcha impossíveis no CE identificado.

Passo 3: Se $J = \emptyset$ *Fim*.

c.c., escolher $\ell \in J$.

Passo 4: Fazer os passos 2 a 4 do algoritmo *Expande Rede* (ponto 2.2).

Passo 5: Alterar o circuito Euleriano – algoritmo *Altera CE* (ponto 3.3).

Passo 6: $J \leftarrow J \setminus \{\ell\}$

Voltar a 3.

Note-se que, em redes de grandes dimensões, deve ser imposto um limite ao número máximo de iterações, pois, em cada iteração a rede vai sendo expandida, dificultando a identificação do novo CE.

Neste capítulo são, em seguida, descritos todos os métodos envolvidos na identificação de um circuito Euleriano que minimize o número de más viragens e que não efectue inversões de marcha que não sejam permitidas.

3.2. *Método de Inspeção*

3.2.1. *Introdução*

Identificar um CE num grafo Euleriano é fácil, difícil é identificar um CE que minimize o número de viragens más em cada cruzamento.

A identificação de um CE foi feita com base no algoritmo apresentado em Gondran e Minoux (1984), mas tendo em conta o tipo de viragem a efectuar em cada nodo.

Nesse algoritmo o CE é construído do seguinte modo:

Dado um grafo Euleriano e um qualquer vértice inicial, seja i , forma-se um qualquer circuito a partir de i para iniciar o CE. O circuito identificado é incluído em CE e retirado do grafo Euleriano. De seguida consideram-se, um a um, os nodos do circuito que ainda tenham arcos neles incidentes (no grafo Euleriano) mas não incluídos em CE. Em cada iteração, toma-se um qualquer desses nodos e identifica-se um novo circuito com arcos do grafo Euleriano que ainda não façam parte do CE. Esse circuito é incluído no CE a partir do nodo em estudo e retirado do grafo Euleriano. O processo é repetido até terem sido incluídos em CE todos os arcos do grafo Euleriano.

A inovação na identificação de um CE, associado ao método de inspecção aqui introduzido é a de escolher, em cada iteração, e para cada nodo, um circuito que tenha em conta o tipo de viragem, em vez de identificar um qualquer circuito. A identificação de um CE, num grafo Euleriano, exige então, o conhecimento do tipo de viragens efectuadas em cada cruzamento (vértice do grafo).

Como se explica no ponto seguinte, neste trabalho, foi criada a matriz de prioridades que guarda toda a informação sobre os tipos de viragens em cada nodo da rede.

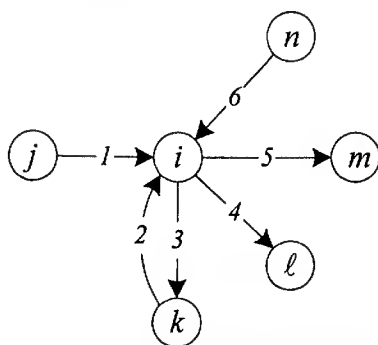
Em 3.2.3 apresenta-se então o algoritmo *Inspecção* de identificação de um CE em que se tente minimizar o número de viragens à esquerda.

3.2.2. Construção da Matriz de Prioridades

A matriz de prioridades, Dir , estabelece o tipo de viragens em cada cruzamento. Assim, a cada nodo da rede associa-se uma linha de Dir de forma a que seja possível identificar o tipo de viragem. O número de elementos não nulos da linha correspondente ao vértice i é igual a $2g(i)$. Considerando-se um arco qualquer com vértice final em i , seja (j, i) , associa-se esse arco ao primeiro elemento da linha i da matriz Dir , considerando $Dir[i, 1] = (j, i)$. De seguida estabelecem-se as prioridades a partir do arco (j, i) , inscrevendo na coluna 2 o arco à direita de (j, i) e assim sucessivamente, até terem sido escritos os $2g(i)$ arcos. No exemplo seguinte ilustra-se a definição de uma linha de Dir .

Exemplo 3.1: Considere-se uma rede em que o número máximo de viragens em todos os nodos é oito, ou seja, em que $cruza = 8$. Sejam os arcos incidentes num vértice i os representados na figura seguinte.

Fig. 3.1: Construção de Dir para o nodo i .



Iniciando com o arco $1 = (j, i)$, a linha i de Dir tem $2g(i) = 6$ elementos não nulos e é igual a: $Dir[i] = [1 \quad 3 \quad 2 \quad 4 \quad 5 \quad 6 \quad 0 \quad 0]$, onde, existindo os dois arcos de sentidos opostos, 2 e 3, se considera sempre primeiro o arco com vértice inicial i .

Assim, se, por exemplo, chegar a i percorrendo o arco $1 = (j, i)$, lendo $Dir[i]$, obtém-se a sequência de $2g(i)$ arcos: 1, 3, 2, 4, 5, 6.

Como i não é extremo inicial de 2 nem de 6, fica a saber-se que relativamente a $1 = (j, i)$:

$3 = (i, k)$ é a rua mais à direita;

$4 = (i, \ell)$ é a segunda rua mais à direita;

$5 = (i, m)$ é a terceira rua mais à direita.

Neste caso, a terceira rua mais à direita pode representar uma viragem à esquerda e a segunda seguir em frente, dependendo da rede.

Se, por outro lado, chegar a i vindo de k , pelo arco $2 = (k, i)$, começa-se a ler a linha i a partir do elemento α : $Dir[i, \alpha] = 2$, em que do último elemento, ou do elemento nulo, se segue para o início da matriz. Resulta então a sequência de $2g(i) = 6$ arcos: 2, 4, 5, 6, 1 e 3, com $\alpha = 3$. Não considerando os arcos 1 e 6 (que não partem de i), relativamente a $2 = (k, i)$, tem-se:

$4 = (i, \ell)$ é a rua mais à direita;

$5 = (i, m)$ é a segunda rua mais à direita;

$3 = (i, k)$ é a terceira rua mais à direita, representando uma inversão de marcha. ♦

A matriz de prioridades pode então ser construída com base no algoritmo seguinte. Embora o conjunto de vértices tenha cardinalidade $n+1$, só se cria a matriz de

prioridades para os vértices de 1 a n , deixando de fora a ETRS (vértice $n + 1$). Estando a estação muito afastada da zona de recolha, as ligações de $n + 1$ aos restantes nodos representam caminhos mais curtos entre os nodos respectivos, não se pondo portanto o problema do tipo de viragens.

Algoritmo: *Cria Dir*

Passo 0: Seja $G_E = (V_E, A_E)$ um grafo Euleriano, com $n + 1 = |V_E|$ e $m = |A_E|$;

$$V_E = \{1, 2, \dots, n, n + 1\};$$

$$\text{Fazer: } \text{cruza} = 2 \times \max_{v \in V_E \setminus \{n+1\}} \{g(v)\};$$

$$\text{Inicializar: } \text{Dir}[i, j]_{i=1, \dots, n, j=1, \dots, \text{cruza}} \leftarrow 0; i \leftarrow 1; j \leftarrow 1;$$

Passo 1: Para o nodo i , considere-se um qualquer arco $a = (k, i) \in A_E$;

$$\text{Fazer: } \text{Dir}[i, j] \leftarrow a;$$

$$b \leftarrow (\text{arco incidente em } i, \text{ mais à direita de } a);$$

$$j \leftarrow j + 1;$$

$$\text{Repetir } \text{Dir}[i, j] \leftarrow b;$$

$$a \leftarrow b;$$

$$\text{Se } (j < 2g(i))$$

$$b \leftarrow (\text{arco incidente em } i, \text{ mais à direita de } a);$$

$$j \leftarrow j + 1$$

$$\text{Até } (j > 2g(i))$$

Passo 2: $j \leftarrow 1; i \leftarrow i + 1$

Se $(i \leq n)$ **voltar a 1**

c.c., FIM.

Note-se que se a rede Euleriana for um multigrafo, os arcos replicados devem ser incluídos em colunas sucessivas de *Dir*. Assim, no passo 1, sempre que se inscreve um arco em *Dir*, há que saber se esse arco tem ou não cópias para que estas possam ser consideradas.

3.2.3. Identificação de um Circuito Euleriano

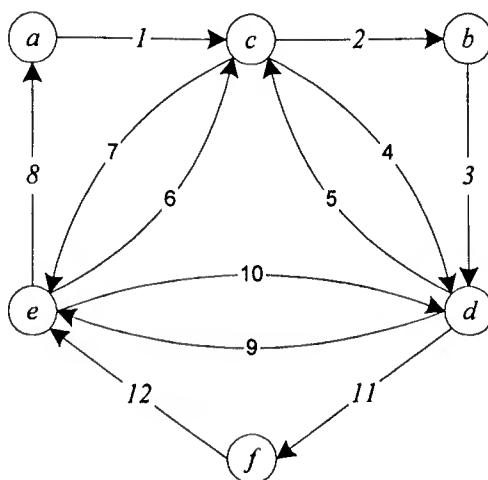
Neste ponto apresenta-se um algoritmo – *Inspecção* – que identifica, num grafo Euleriano, um circuito Euleriano, CE, tendo em conta o tipo de viragens.

Identificadas na matriz *Dir* os tipos de viragens em cada nodo, vai construir-se um CE em que se tenta minimizar o número de más viragens. Tendo que incluir todos os arcos do grafo em CE e, em particular, todos os arcos com extremo final e inicial em cada um dos nodos do grafo, põe-se a questão da escolha do arco que deve ser usado em primeiro lugar, a partir de um qualquer vértice. Ou seja, para cada nodo com grau superior a um, qual o arco que deve ser escolhido em primeiro lugar para incluir em CE, o mais à esquerda ou o mais à direita?

No exemplo que se segue é ilustrada a identificação de um CE, de acordo com o algoritmo de Gondran e Minoux (1984). De seguida apresenta-se a alteração para o algoritmo implementado nesta tese, sendo justificada a regra específica para a escolha de cada arco a incluir em CE, em cada passo do algoritmo.

Exemplo 3.2: Considere-se o grafo Euleriano seguinte, com 6 nodos e 12 arcos.

Fig. 3.2: Grafo Euleriano, *G*.



Considere-se um circuito qualquer que inclua o vértice a , como sendo o CE inicial.

CE	Lista de arcos de G por considerar em CE
$CE = (a, c, e, a) = (1, 7, 8)$	$[2, 3, 4, 5, 6, 9, 10, 11, 12]$

Não existe em G mais nenhum circuito que inclua a , pois $g(a) = 1$. Assim, analisam-se os restantes vértices no circuito. O vértice c , segundo vértice em CE , tem grau 3, e ainda só foi considerado um circuito que o contém. Sendo assim, outro circuito em c , com arcos ainda não considerados é: $(c, d, f, e, c) = (4, 11, 12, 6)$. Este circuito é então incluído entre os arcos 1 e 7, resultando:

CE	Por considerar
$(a, c, d, f, e, c, e, a) = (1, 4, 11, 12, 6, 7, 8)$	$[2, 3, 5, 9, 10]$

Em G , ainda se pode formar outro circuito que inclua c , com arcos ainda não considerados, que é: $(c, b, d, e, d, c) = (2, 3, 9, 10, 5)$. Incluindo entre o arco 1 e o arco 4, obtém-se o circuito Euleriano final:

$$(a, c, b, d, e, d, c, d, f, e, c, e, a) = (1, 2, 3, 9, 10, 5, 4, 11, 12, 6, 7, 8).$$
♦

Com este método, em que não se pondera qual o melhor arco para virar em cada nodo, obtiveram-se, e sem considerar as sequências de arcos obrigatórias, (2,3) e (11,12); uma viragem para a rua mais à direita, 7-8, duas em frente, 3-9 e 12-6, duas viragens à esquerda, 1-2 e 10-5 e três inversões de marcha, 9-10, 5-4 e 6-7. Num total de cinco más viragens.

Ao gerar um CE para o problema em estudo pretende-se que as viragens se efectuem da forma mais acessível e permitida pelas regras de trânsito, ou seja, pretende-se evitar as viragens à esquerda e inversões de marcha.

Várias hipóteses surgem para tentar impedir as más viragens.

Sendo mais fácil começar com um circuito pequeno que vai sendo alargado, até ficar completo, surge a ideia de fazer as viragens começando por escolher o arco mais à direita possível. Contudo, no algoritmo de construção de CE a primeira escolha é muitas vezes alterada, pois, subcircuitos vão sendo incluídos no circuito em construção.

Por outro lado, escolher a pior viragem sempre em primeiro lugar, para posteriormente ser melhorada, pode ser perigoso, pois, muitas vezes, a primeira escolha não é alterada.

Este tipo de escolha, inicialmente sugerida por Male e Liebman (1978), foi também utilizada por Eglese e Li (1992).

Nesta tese, optou-se por escolher, como próximo arco a incluir em CE, o arco que represente a segunda melhor viragem, que, sendo por vezes melhorada, não representa uma muito má escolha quando não sofre alteração.

Fica no entanto, como pesquisa futura, a comparação entre diversos critérios de escolha, considerando uma escolha aleatória como um dos critérios possíveis.

Retomando o exemplo anterior, e seguindo o critério da segunda melhor viragem, resulta a sequência que se ilustra de seguida.

Exemplo 3.3: Considere-se de novo o grafo da figura 3.2 e construa-se o CE seguindo o critério de escolha da segunda melhor viragem em cada caso. Uma vez que é necessário conhecer o tipo de viragem em cada nodo, há que criar a matriz *Dir*. Da rede com 6 nodos e $\max_{v \in V} \{g(v)\} = 3$, resulta $cruza = 6$ e *Dir*, de dimensão 6×6 , é:

$$Dir = \begin{matrix} (a) \\ (b) \\ (c) \\ (d) \\ (e) \\ (f) \end{matrix} \begin{bmatrix} 8 & 1 & 0 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 1 & 7 & 6 & 4 & 5 & 2 \\ 3 & 5 & 4 & 9 & 10 & 11 \\ 7 & 8 & 12 & 10 & 9 & 6 \\ 11 & 12 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Partindo do nodo a , e como é o início, escolhe-se para fazer parte de CE o primeiro arco da linha a de Dir que tenha início em a . Ou seja, $l = Dir[a, 2]$, será o primeiro arco de CE, $CE = (a, c) = (l)$.

De seguida, na linha c de Dir , escolhe-se, se possível, o segundo arco mais à direita do arco l . Para tal, identifica-se $k : Dir[c, k] = l$. Neste caso, $k = 1$. Como $Dir[c, k + 2] = 6$ e $6 = (e, c)$, não tem início em c , escolhe-se o arco mais à direita deste para continuar o CE, ou seja, $Dir[c, 4] = 4$. Logo, $CE = (a, c, d) = (l, 4)$.

Lendo a linha d de Dir , observa-se, seguindo o mesmo método, que o segundo arco mais à direita de 4 com o sentido (d, \bullet) é $11 = (d, f)$. Incluindo em CE este arco, vem: $CE = (a, c, d, f) = (l, 4, 11)$. Prosseguindo desta forma, até voltar ao vértice de partida, obtém-se:

$$CE = (a, c, d, f, e, c, b, d, e, a) = (1, 4, 11, 12, 6, 2, 3, 9, 8).$$

Construído um circuito, analisam-se, a partir do início, todos os vértices de CE para ver se ainda existem arcos da rede por considerar em CE.

Existindo ainda arcos com início em c , forma-se um novo circuito para incluir a partir do nodo c , em CE, escolhendo sempre que possível, o segundo arco mais à direita do presente arco.

O subcircuito que será considerado a partir de c , será então: $(7, 10, 5)$ e $CE = (a, c, e, d, c, d, f, e, c, b, d, e, a) = (1, 7, 10, 5, 4, 11, 12, 6, 2, 3, 9, 8)$.

Com este CE, realiza-se: uma viragem à direita, $1-7$; cinco em frente, $4-11$, $12-6$, $6-2$, $3-9$ e $9-8$, duas viragens à esquerda, $7-10$ e $10-5$ e apenas uma inversão de marcha, $5-4$, perfazendo três más viragens. ♦

Assim, no algoritmo seguinte, implementado para a identificação do CE com base no algoritmo de Gondran e Minoux (1984), é escolhido, sempre que possível, o segundo arco mais à direita como próximo arco a incorporar no circuito. Para tal denomine-se por:

- $CE[\bullet]$ um vector de dimensão $m+1$, que representa a sequência de arcos do circuito, onde,

se $a = CE[u]$, o arco a segue u no Circuito Euleriano,

$u_1 = CE[m+1]$ é o primeiro arco de CE,

$\alpha : CE[\alpha] = -1$ representa o último arco de CE.

- x o vértice inicial.

Algoritmo: *Inspecção*

Passo 0. Construir a matriz de prioridades de G_E : algoritmo *Cria Dir*

Escolher $x \in V$ um qualquer nodo para início

Fazer: $\bar{A} \leftarrow A_E$; $CE[u] \leftarrow 0$, $u = 1, 2, \dots, m$; $CE[m+1] \leftarrow -1$

$inicio \leftarrow x$ {nodo inicial dos subcircuitos}

$conta \leftarrow 0$ {n.º de arcos inseridos em CE}

$vira \leftarrow 2$ {indica o arco para onde deve virar}

$aux \leftarrow -1$ {fim do circuito a inserir em CE}

$seg \leftarrow x$

$a \leftarrow Dir[x, vira]$

Passo 1. Se $a = 0$ ir para 9

Se ($a \in \bar{A}$ e $a > 0$ e a tem início em seg) fazer: $u \leftarrow a$

c.c., $u \leftarrow 0$

Se $(vira = 3)$ fazer: $vira \leftarrow 2$; $a \leftarrow Dir[x, vira]$

c.c., Se $(vira = 2)$ fazer: $vira \leftarrow 4$; $a \leftarrow Dir[x, vira]$

c.c., $vira \leftarrow vira + 1$; $a \leftarrow Dir[x, vira]$

Passo 2. Se $(u = 0)$ voltar a 1,

Passo 3. Fazer: $\bar{A} \leftarrow \bar{A} \setminus \{u\}$ e considerar que $u = (u_1, u_2)$

$conta \leftarrow conta + 1$

$aux \leftarrow CE[m + 1]$

$CE[m + 1] \leftarrow u$

$CE[u] \leftarrow aux$

$ant \leftarrow u_1$

$arco_ant \leftarrow u$

Passo 4. Procurar na linha u_2 de Dir $arco_ant$, seja $Dir[u_2, j] \leftarrow arco_ant$

Fazer $vira \leftarrow j + 2$

Passo 5. Se $vira > cruza$ então $vira \leftarrow 2$

$a \leftarrow Dir[u_2, vira]$

Se $(a \in \bar{A}$ e $a > 0$ e a tem início em u_2) então: $u = (u_1, u_2) \leftarrow a$; $seg \leftarrow u_2$

c.c., Se $(vira = 1)$ fazer $vira \leftarrow cruza$

c.c., $vira \leftarrow vira + 1$

voltar a 5

Passo 6. Fazer: $\bar{A} \leftarrow \bar{A} \setminus \{u\}$

$conta \leftarrow conta + 1$

$aux \leftarrow CE[arc_ant]$

$CE[arc_ant] \leftarrow u$

$CE[u] \leftarrow aux$

$ant \leftarrow u_1$

$arco_ant \leftarrow u$

Passo 7. **Se** $(seg = x)$ **ir para 8**

c.c., Se $(seg = inicio)$ **ir para 9**

c.c., voltar a 4

Passo 8. $vira \leftarrow 3$

$a \leftarrow Dir[x, vira]$

voltar a 1

Passo 9. **Se** $(conta = m)$ **então FIM**

c.c., Encontrar no circuito em construção um nodo, a_1 , inicial

 de um arco ainda não considerado, seja $a = (a_1, a_2) \in \bar{A}$

 Seja $arco_ant = (u_1, u_2)$ o arco antecessor de a em CE

 e $inicio = u_2$

Voltar a 4.

3.3. *Alteração do Circuito Euleriano*

O Circuito Euleriano obtido com o algoritmo *Inspecção* evita as más viragens mas não as impede. Regra geral, em cada rede existem inversões de marcha não permitidas que podem não ser impedidas no CE identificado pelo algoritmo *Inspecção*. Como se viu, neste algoritmo, embora se tenha em atenção o tipo de viragens, nem sempre se conseguem impedir certas inversões de marcha. Por outro lado, expandir inicialmente a rede para que não seja possível efectuar inversões de marcha não permitidas pode aumentar muito as dimensões do problema. Assim, optou-se por iniciar a identificação do CE tendo cuidado com o tipo de viragens e, caso tal origine inversões não permitidas, a rede é expandida nos vértices onde essas inversões são realizadas sendo, de seguida, identificado um novo CE.

A expansão da rede é então feita vértice a vértice onde existem inversões não permitidas. De acordo com o algoritmo *Identifica CE*, descrito em 3.1, é identificado um primeiro CE tendo em conta as más viragens. Se existir uma sequência de arcos, em CE, que represente uma inversão de marcha não permitida num certo vértice, é expandida a rede, nesse nodo, através do algoritmo *Expande Rede*. De seguida é identificado um novo circuito Euleriano – algoritmo *Altera CE* – de acordo com os novos arcos da rede expandida.

O algoritmo *Altera CE* começa por eliminar de CE todos os arcos que envolvem o nodo onde não é permitido inverter, seja $\ell \in J$. O CE é então percorrido até encontrar uma interrupção na sequência de arcos. Quando tal acontece, é identificado um caminho na rede expandida até reencontrar a sequência do CE já construído. Quando o CE estiver de novo completo verifica-se se existem subcircuitos isolados em vez de um só circuito,

caso em que se incorporam, um a um, no CE maior, tendo atenção para não efectuar novamente esta inversão de marcha ou outra que já tivesse sido anteriormente eliminada.

Algoritmo: *Altera CE*

Passo 0. Seja $\ell \in J$ o vértice expandido e $x \in V$ o vértice inicial de CE

Passo 1. Retirar de CE todos os arcos incidentes em ℓ , fazendo

$$CE[u] \leftarrow 0, \forall u \text{ incidente em } \ell$$

$$v \leftarrow CE[m+1]$$

$$conta \leftarrow 1 \quad \{\text{n.º de arcos já analisados}\}$$

Passo 2. Se $v = 0$, Fazer: $e \leftarrow (x, \ell)$, $v \leftarrow e$, $CE[m+1] \leftarrow v$ e ir para 3

$$\text{c.c. } u \leftarrow CE[v] \text{ e ir para 4}$$

Passo 3. {Sair de ℓ }

Procurar um arco da rede expandida sucessor de v , seja $s = (\ell, s_2)$.

$$\text{Fazer: } u \leftarrow s$$

$$CE[v] \leftarrow u$$

$$conta \leftarrow conta + 1$$

Passo 4. Se $(u = 0)$ ir para 5

$$\text{c.c., } v \leftarrow u$$

$$\text{Se } (conta < m)$$

$$\text{Se } (u \neq -1)$$

$$\text{Fazer } u \leftarrow CE[v]$$

$$conta \leftarrow conta + 1$$

voltar a 4

c.c., ir para 6

c.c., ir para 7

Passo 5. {Entrar em ℓ }

Procurar um arco da rede expandida sucessor de v e com vértice final em ℓ ,

seja $e = (v_2, \ell)$.

Fazer $u \leftarrow e$

$CE[v] \leftarrow u$

$conta \leftarrow conta + 1$

$v \leftarrow u$

Se $CE[v] = 0$ voltar a 3

Passo 6. **Se $\exists v : CE[v] = 0, v = 1, \dots, m+1$**

Procurar um arco da rede expandida sucessor de v com vértice final

ou inicial em ℓ , seja $e = (v_1, \ell)$ ou $s = (\ell, s_2)$,

Fazer $u \leftarrow e$ ou $u \leftarrow s$

$CE[v] \leftarrow u$

$conta \leftarrow conta + 1$

Se $u = e$ Fazer $v \leftarrow u$ e voltar a 3

c.c., voltar a 4

Passo 7. {Análise à existência de subcircuitos}

$u \leftarrow CE[m+1]$

$conta \leftarrow 1$

Passo 8. $v \leftarrow CE[u]$

Se $v \neq -1$

Fazer $conta \leftarrow conta + 1$

$u \leftarrow v$

voltar a 8

Passo 9. **Se** $conta = m$ **FIM**

c.c.,

Procurar um subcircuito e inseri-lo em CE de forma a não fazer uma inversão de marcha proibida.

Se tal não for possível então

“Não é possível evitar uma inversão de marcha proibida”

inserir o subcircuito em CE

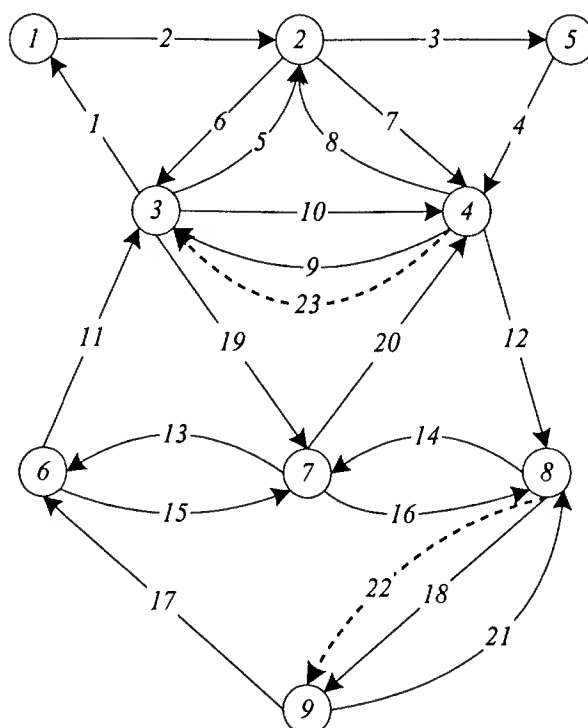
Voltar a 7.

No ponto seguinte apresenta-se um exemplo com o objectivo de ilustrar os algoritmos desenvolvidos e implementados.

3.4. Ilustração

Neste ponto ilustra-se o funcionamento dos algoritmos propostos utilizando um exemplo de uma rede Euleriana com 9 nodos e 23 arcos.

Fig. 3.3 – Rede Euleriana



Neste multigrafo Euleriano 6 ruas têm dois sentidos e 9 ruas têm sentido único, existindo ainda 2 ruas que irão ser percorridas em vazio, sem efectuar recolha, e uma inversão de marcha que não é possível evitar, pois, inevitavelmente, uma das inversões (18, 21) ou (22, 21) terá que ser feita.

Para obter um circuito Euleriano que tenha em conta as más viragens e que não faça inversões de marcha, é necessário primeiro identificar um circuito Euleriano que apenas minimize o número de más viragens (viragens à esquerda e inversões de marcha), algoritmo *Inspecção* para posteriormente impedir apenas as inversões de marcha que são efectuadas, algoritmos *Expande Rede* e *Altera CE*.

O algoritmo *Inspecção* inicia com a construção da matriz de prioridades, *Dir* de dimensão 9×8 , pois, $cruza = 2 \max_{v \in V} \{g(v)\} = 2g(3) = 8$ e $nodos = 9$.

$$Dir = \begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 6 & 5 & 7 & 8 & 3 & 0 & 0 \\ 6 & 1 & 11 & 19 & 10 & 9 & 23 & 5 \\ 4 & 8 & 7 & 9 & 23 & 10 & 20 & 12 \\ 3 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 17 & 15 & 13 & 11 & 0 & 0 & 0 & 0 \\ 19 & 13 & 15 & 16 & 14 & 20 & 0 & 0 \\ 12 & 14 & 16 & 18 & 22 & 21 & 0 & 0 \\ 18 & 22 & 17 & 21 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Para o algoritmo *Inspecção* considera-se que o vértice inicial é o vértice 1 e o número de arcos é 23.

De seguida vão-se ilustrar todos os passos efectuados na construção do CE inicial. Para uma leitura mais fácil da construção de CE, representa-se por CE_i o vector que representa a sequência de arcos que se inserem em CE.

Começando no nodo 1, e uma vez que é o início do circuito (*Passo 0*), escolhe-se para fazer parte de CE_i o primeiro arco da linha 1 de *Dir*, que tenha início no nodo 1.

Sendo assim, $arco\ inicial = Dir[1, 2] = 2$, será o primeiro arco de CE_i (*Passo 3*),

$$CE_i = [1, 2] = (2).$$

De seguida, na linha 2 de *Dir*, escolhe-se, se possível, o segundo arco mais à direita do arco 2, (*Passo 4*), ou seja, $Dir[2, 3] = 5$. Como este arco não tem início no nodo 2 (*Passo 5*), escolhe-se o arco seguinte à direita deste para continuar o CE, ou seja, $arco\ seguinte = Dir[2, 4] = 7$.

Logo, (**Passo 6**),

$$CE_i = [1, 2, 4] = (2, 7).$$

Passando para o nodo 4, e lendo a linha 4 de *Dir*, (**Passo 4**), verifica-se que o arco seguinte, de acordo com a mesma metodologia (**Passo 5**) é o arco 21. Incluindo em CE_i este arco, fica (**Passo 6**):

$$CE_i = [1, 2, 4, 3] = (2, 7, 23).$$

Note-se que o arco 23, corresponde à 1ª rua mais à direita, no entanto foi considerado, pela leitura da matriz *Dir*, como o 2º arco mais à direita por se tratar de uma réplica do arco 9.

Prosseguindo desta forma, até voltar ao nodo de partida, (**Passo 7**), obtém-se:

$$CE_i = [1, 2, 4, 3, 1] = (2, 7, 23, 1).$$

$\uparrow \quad \nearrow \quad \uparrow$

Analisando as viragens efectuadas neste circuito inicial, obtiveram-se duas viragens em frente, assinaladas com \uparrow , e uma à direita, assinalada por \nearrow .

Como já não existem mais arcos incidentes no nodo inicial, (**Passo 8 e Passo 1**), nodo 1, passa-se para o nodo seguinte do CE_i , (**Passo 9**), ou seja, para o nodo 2.

Neste nodo inicia-se um novo circuito, $CE1$, analogamente ao caso anterior. Este novo circuito é depois inserido no CE_i a seguir ao arco 2.

Obtém-se, assim o subcircuito,

$$CE1 = (3, 4, 9, 5),$$

que se insere em CE_i , após o arco 2, resultando,

$$CE_i = (2, 3, 4, 9, 5, 7, 23, 1).$$

$\nwarrow \quad \uparrow \quad \nearrow \quad \nearrow \quad \nearrow \quad \uparrow$

Como no nodo 2 a viragem passou a ser para a esquerda, \nwarrow , o circuito passou a ter uma viragem à esquerda, duas em frente e três à direita.

A sequência de arcos (3, 4) não é classificada uma vez que é a única possível para o respectivo nodo.

No nodo 2 ainda existe um arco incidente. A viragem vai então efectuar-se para o primeiro arco à direita. Obtém-se assim um novo subcircuito, CE_2 ,

$$CE_2 = (6, 19, 16, 22, 21, 14, 20, 8),$$

que se insere em CE_i , novamente depois do arco 2, resultando,

$$CE_i = (2, \underset{\nearrow}{6}, \underset{\uparrow}{19}, \underset{\uparrow}{16}, \underset{\nearrow}{22}, \underset{\curvearrowright}{21}, \underset{\nearrow}{14}, \underset{\nearrow}{20}, \underset{\uparrow}{8}, 3, 4, 9, 5, 7, 23, 1).$$

Neste caso a viragem efectuada no nodo 7, 14-20, não se efectuou para o 2º arco porque na matriz Dir , tem-se $Dir[7, k + vira] = 0$, logo, considerando $vira = 1$, obtém-se $Dir[7, k + 1] = 20$, o que corresponde ao 1º arco à direita.

Este circuito melhorou a viragem no nodo 2, dado que se eliminou a viragem à esquerda efectuada anteriormente e surgiu uma inversão de marcha, \curvearrowright , que não se pode evitar, como foi referido anteriormente. O circuito ficou então com uma inversão de marcha, cinco viragens em frente e oito viragens à direita.

Passando para o nodo seguinte do circuito, nodo 3, obtém-se então o subcircuito CE_3 ,

$$CE_3 = (10, 12, 18, 17, 11),$$

Este subcircuito é agora inserido a seguir do nodo 3, arco 6,

$$CE_i = (2, \underset{\nearrow}{6}, \underset{\nwarrow}{10}, \underset{\nwarrow}{12}, \underset{\nwarrow}{18}, \underset{\nwarrow}{17}, \underset{\nwarrow}{11}, \underset{\uparrow}{19}, \underset{\uparrow}{16}, \underset{\nearrow}{22}, \underset{\curvearrowright}{21}, \underset{\nearrow}{14}, \underset{\nearrow}{20}, \underset{\uparrow}{8}, 3, 4, 9, 5, 7, 23, 1).$$

Neste circuito surgiram três viragens à esquerda, sendo uma delas no nodo onde se inseriu o novo circuito. Existem agora, neste circuito, uma inversão de marcha, três viragens à esquerda, quatro em frente e onze à direita.

Prosseguindo desta forma, o Circuito Euleriano obtido com o algoritmo *Inspecção* é:

$$CE_i = (2, 6, 10, 12, 18, 17, 15, 13, 11, 19, 16, 22, 21, 14, 20, 8, 3, 4, 9, 5, 7, 23, 1).$$

$\nearrow \nwarrow \nearrow \nwarrow \nearrow \nearrow \curvearrowright \nearrow \nearrow \uparrow \nearrow \curvearrowright \nearrow \nearrow \uparrow \nearrow \uparrow \nearrow \nearrow \nearrow \uparrow$

Foi eliminada uma viragem à esquerda, no entanto surgiu uma inversão de marcha, no nodo 7, sequência de arcos (15, 13).

Tabela 3.1 - Resumo das viragens efectuadas em CE_i .

<i>Direita</i>	<i>Frente</i>	<i>Esquerda</i>	<i>Inversão</i>
<i>13</i>	<i>4</i>	<i>2</i>	<i>2</i>

A primeira má viragem que se efectua verifica-se na sequência de arcos (6, 10). Esta manobra representa uma viragem à esquerda e equivale, como se explica de seguida, à segunda viragem à direita possível quando se vem do arco 6.

Neste nodo, nodo 3, a viragem mais à direita seria para o arco 1, no entanto, tal viragem fecharia o circuito. Assim a viragem para o arco 19 é efectivamente a primeira hipótese e a viragem efectuada, e a sequência (6, 10) é portanto a segunda hipótese de viragem.

O mesmo se passa para a segunda viragem à esquerda efectuada, sequência de arcos (12, 18), que também representa a segunda hipótese de viragem quando se vem do arco 12.

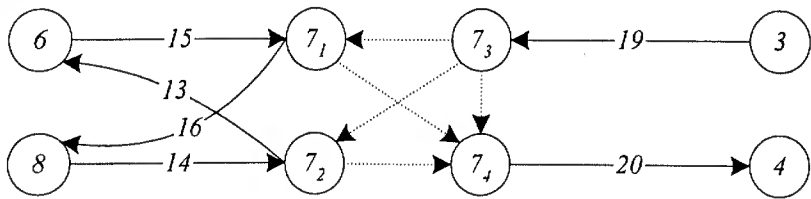
Em relação às duas inversões de marcha efectuadas, apenas uma delas é possível proibir, a inversão de marcha no nodo 7, com a sequência de arcos (15, 13). Como facilmente se pode ver na figura 3.3, a inversão efectuada no nodo 9 não se pode proibir.

Sendo assim, vai-se proibir a inversão em 7. Como neste nodo existe outra sequência de arcos que poderá vir a originar uma inversão de marcha, (14, 16), impõe-se logo a proibição de duas inversões de marcha no nodo 7.



Passa-se assim para o algoritmo *Expande Rede*, que vai expandir o nodo 7, do seguinte modo:

Fig. 3.4 – Nodo 7 expandido



Após a expansão da rede altera-se CE_i , algoritmo *Altera CE*, de acordo com os novos caminhos que envolvem o nodo 7.

Este algoritmo começa por eliminar de CE_i todos os arcos incidentes em 7, (*Passo 1*).

$$CE_a = (2, 6, 10, 12, 18, 17, 0, 0, 11, 0, 0, 22, 21, 0, 0, 8, 3, 4, 9, 5, 7, 23, 1).$$

Percorre-se então o circuito até encontrar uma interrupção. Esta surge depois do arco $17 = (9, 6)$. Insere-se no circuito um arco entre o último nodo antes da interrupção, 6, e o nodo expandido, nodo 7, ou seja, insere-se o arco $15 = (6, 7)$ (*Passo 5*).

De acordo com a rede expandida, existem dois arcos que poderiam suceder a $15 = (6, 7_1)$, o arco $16 = (7_1, 8)$ e o arco $20 = (7_1, 7_4, 4)$. Quando tal acontece é sempre utilizado primeiro aquele que passe apenas por um dos nodos expandidos, ou seja, é utilizado o arco 16, (*Passo 3*).

Para não modificar muito CE_i , apenas são alteradas as viragens efectuadas no nodo 7. Sendo assim, ao incluir em CE_a a sequência de arcos $(17, 15, 16)$, como 22 não foi eliminado e é o arco sucessor de 16 em CE_i , a sequência de arcos $(17, 15, 16, 22)$ substitui a interrupção detectada em CE_a , (*Passo 4*).

Note-se que com esta sequência de arcos o algoritmo *Altera CE*, deixa para trás o arco 11 e continua a pesquisa de interrupções. Este arco surgirá com outra sequência de arcos a alterar no nodo expandido.

Prossegue-se o circuito de acordo com o último arco da sequência atrás referida.

$$CE_a = (2, 6, 10, 12, 18, 17, 15, 16, 22, 21, 0, 0, 8, 3, 4, 9, 5, 7, 23, 1).$$

Continuando a percorrer o circuito até encontrar nova interrupção, (*Passo 4*), esta verifica-se a seguir ao arco 21. Seguindo novamente a rede expandida, (*Passos 5 e 3*) surge a sequência de arcos (21, 14, 13, 11). Inserindo esta sequência em CE_a ,

$$CE_a = (2, 6, 10, 12, 18, 17, 15, 16, 22, 21, 14, 13, 11, 0, 0, 8, 3, 4, 9, 5, 7, 23, 1).$$

A sequência de arcos que segue o arco 11, último arco antes da interrupção detectada, (*Passo 5 e 3*) é a sequência (11, 19, 20, 8).

Utilizados todos os arcos incidentes no nodo 7 expandido, completa-se o circuito Euleriano, agora CE_f , sem inversões de marcha no nodo 7.

$$CE_f = (2, 6, 10, 12, 18, 17, 15, 16, 22, 21, 14, 13, 11, 19, 20, 8, 3, 4, 9, 5, 7, 23, 1)$$

↗↖↗↖↗↗↗↗↻↗↖↗↗↖↑↗↑↗↗↑

No nodo 7 deixou de existir inversão de marcha, mas, passaram a efectuar-se duas viragens à esquerda.

Tabela 3.2 – Resumo das viragens efectuadas em CE_f .

<i>Direita</i>	<i>Frente</i>	<i>Esquerda</i>	<i>Inversão</i>
<i>13</i>	<i>3</i>	<i>4</i>	<i>1</i>

O CE obtido em CE_f não contém subcircuitos, (*Passo 7 e 8*), obtendo-se assim um CE (*Passo 9*) sem inversões de marcha consideradas proibidas.

Capítulo 4. Resultados Computacionais

Neste capítulo analisam-se os algoritmos propostos em função de resultados computacionais obtidos num conjunto de problemas teste.

Os algoritmos foram implementados em Pascal, tendo sido utilizado o Software Borland Delphi, versão 2.0, num processador *Pentium* a 450MHz, para obter os resultados que se apresentam e analisam.

A geração dos problemas teste, teve como objectivo a obtenção de redes que possam ser consideradas representativas de redes de estradas urbanas.

Esta geração pode ser considerada dividida em 4 etapas. Primeiro é gerado um grafo não orientado, planar e conexo. Esse grafo é transformado num grafo orientado, planar e fortemente conexo, sendo aleatoriamente gerados os dados associados, ou seja, os custos de serviço e em vazio, bem como as procuras. A resolução de um problema de Transportes, anteriormente referida, permite então a transformação do grafo de procuras, gerado aleatoriamente, num grafo Euleriano (Mourão (1997)).

O objectivo deste estudo é o de analisar o número de más viragens efectuadas na rota de cada veículo quando utilizado o método heurístico aqui proposto, antes e depois da expansão das redes originada pelas inversões de marcha proibidas.

Dado que as redes geradas são representativas de pequenas zonas de uma cidade, existem ruas com sentido único e ruas com dois sentidos, cruzamentos com várias ruas e ainda becos. Como é óbvio não se irá contabilizar o número de inversões de marcha

efectuadas em becos nem o número de inversões de marcha efectuadas em ruas onde estas não são evitáveis, mesmo não se tratando de becos, como se ilustrou na figura 3.3. Foram geradas aleatoriamente redes com 17, 26, 37 e 50 nodos. Cada problema teste representa a média de cinco redes, com as mesmas características, ou seja, com o mesmo número de nodos e aproximadamente o mesmo número de arcos. Apresentam-se os resultados obtidos para 8 problemas teste de cada grupo de nodos.

No grafo Euleriano obtido como descrito anteriormente é identificado um Circuito Euleriano que tenha em conta o tipo de viragens - algoritmo *Inspecção*. Contabilizado o número de más viragens, viragens à esquerda, inversões de marcha e o número de cruzamentos onde se efectuam essas inversões de marcha, é obtida a rede expandida, algoritmo *Expande Rede*, nos nodos onde se pretende proibir as inversões de marcha. Para analisar o número de inversões de marcha que são evitadas, optou-se por proibir todas as inversões efectuadas pelo algoritmo *Inspecção*. Finalmente é obtido um novo Circuito Euleriano, com o algoritmo *Altera CE*, de acordo com a nova rede, onde são evitadas as inversões de marcha proibidas, sempre que tal seja possível.

Os resultados obtidos nos diferentes problemas teste são apresentados em três tabelas, pois estando o método heurístico dividido em três fases distintas toma-se conveniente a análise dos resultados de cada uma destas fases.

Na primeira tabela, tabela 4.1, encontram-se os resultados do algoritmo *Inspecção* nas redes com 17 nodos. Para cada problema teste é apresentado o intervalo de arcos das cinco redes geradas, 2ª coluna. Num grafo com n arcos realizam-se $n-1$ sequências de arcos que representam viragens. Para melhor analisar o número de más viragens apresenta-se na 3ª coluna o número médio de viragens efectuadas; na 4ª coluna, o número médio de viragens à esquerda; na 5ª coluna, o número médio de inversões de marcha; na última coluna, a percentagem de más viragens efectuadas. Nas últimas

linhas da tabela encontram-se os valores médios dos oito problemas teste, bem como os valores mínimos e máximos.

Tabela 4.1 – Resultados do algoritmo *Inspecção* em redes de 17 nodos.

Problema teste	N.º de arcos	N.º médio de viragens	N.º médio de		Total de más viragens (%)
			viragens à esquerda	inversões de marcha	
1	31, 32	31	2	2,2	13,5
2	33, 34	33	2,8	2,8	17
3	35	34	2,2	3	15,3
4	36	35	3,8	5,2	25,7
5	37, 38	37	2,2	2,4	12,4
6	39, 40	38	2,4	2,4	12,6
7	41	40	3	4	17,5
8	43, 47	44	4,8	2,4	16,4
Média	—	—	2,9	3,1	16,4
Mínimo	31	78	2	2,2	12,4
Máximo	47	103	4,8	5,2	25,7

Da observação desta tabela verifica-se que o número de arcos varia entre 31 e 47 para as redes geradas com 17 nodos. O algoritmo *Inspecção* originou circuitos com cerca de 16,4% de más viragens, das quais 48,3% são do tipo de viragens para a rua mais à esquerda e, a maior parte, 51,7%, são inversões de marcha.

Com o objectivo de comparar as viragens antes e depois da expansão da rede, na tabela 4.2 encontram-se os resultados relativos ao número de viragens à esquerda efectuadas com o algoritmo *Inspecção* (3ª coluna) e após a expansão da rede e respectiva alteração do circuito (4ª coluna), algoritmos *Expande Rede* e *Altera CE*. Na última coluna apresenta-se a percentagem da redução verificada neste tipo de viragens.

Tabela 4.2 – Viragens à esquerda em problemas com 17 nodos

Problema teste	N.º de arcos	Viragens à esquerda		Redução das vir. esq. (%)
		Alg. Inspeção	Rede Expandida	
1	31, 32	2	0,8	60
2	33, 34	2,8	2,2	21,4
3	35	2,2	2	9,1
4	36	3,8	2,8	26,3
5	37, 38	2,2	1,8	18,2
6	39, 40	2,4	2	16,7
7	41	3	1,6	46,7
8	43, 47	4,8	2,8	41,7
<i>Média</i>	—	2,9	2	30
<i>Mínimo</i>	31	2	0,8	9,1
<i>Máximo</i>	47	4,8	2,8	60

Como se pode observar, o número de viragens à esquerda diminuiu sempre e significativamente com a expansão da rede. Verificou-se, em média, uma diminuição de 30% das viragens à esquerda, atingindo um valor máximo de 60% de redução do número de viragens à esquerda.

É de referir que, no algoritmo *Altera CE*, nada é feito para que sejam melhoradas as viragens à esquerda. Porém, o impedimento de certas inversões de marcha origina alteração de diversas viragens provocando, em particular, uma diminuição das viragens à esquerda passando a efectuar-se mais viragens à direita do que anteriormente.

Na tabela 4.3 apresentam-se os resultados relevantes à análise ao número de inversões de marcha. Assim, na 2ª coluna consta o número de inversões de marcha nos circuitos

de cada problema teste; na 3ª, o número de nodos onde se efectuam essas inversões; na coluna 4, encontra-se o número de nodos que foram expandidos (como referido pretende-se proibir todas as inversões de marcha efectuadas no circuito); na 5ª coluna detalha-se o número de inversões de marcha que não foi possível evitar, entre as efectuadas inicialmente; na 6ª coluna o número de novas inversões de marcha que surgiram com as proibições impostas às primeiras; e, por último, o número de nodos onde existem inversões de marcha no circuito final, depois da rede expandida e de alterado o circuito.

Tabela 4.3 – Inversões de marcha em problemas com 17 nodos.

Problema teste	Algoritmo <i>Inspecção</i>		N.º de nodos expandidos	Algoritmo <i>Altera CE</i>		
	N.º de inversões de marcha	N.º de nodos com inversões		N.º de inversões mantidas novas		N.º de nodos com inversões
1	2,2	2	1,4	0	0,4	0,4
2	2,8	2	2	0	1,8	1,8
3	3	2	1,8	0,2	1	1,2
4	5,2	4,2	4	0,4	2,6	2,2
5	2,4	2,2	2	0,4	1,4	1,8
6	2,4	2,2	2	0,4	0,8	1
7	4	4	4	0,2	2,4	2
8	2,4	2,2	2,2	0	2,2	1,4
<i>Média</i>	3,1	2,6	2,4	0,2	1,6	1,5
<i>Mínimo</i>	2,2	2	1,4	0	0,4	0,4
<i>Máximo</i>	5,2	4,2	4	0,4	2,6	2,2

Observa-se, da tabela 4.3, que foram expandidos 92% dos nodos onde se realizavam inversões de marcha. A não expansão da totalidade destes nodos pode dever-se a que, à

medida que a rede vai sendo expandida, o CE vai sendo alterado e, consequentemente, são modificadas certas viragens nos restantes nodos, em particular, algumas inversões de marcha podem deixar de se efectuar, deixando de ser preciso expandir os respectivos nodos.

Não foram eliminadas a totalidade das inversões de marcha que se pretendiam proibir, tendo-se mantido 6,5%. Tal pode dever-se ao facto de não ser possível proibir inversões de marcha expandindo apenas o nodo onde ela é proibida, ou seja, pode ser necessário analisar uma zona de nodos. Obviamente que casos há em que não é mesmo possível proibir a inversão de marcha, pois tal proibição iria impossibilitar a construção de um circuito Euleriano.

A eliminação de certas inversões de marcha pode também fazer com que novas inversões tenham que se efectuar em nodos vizinhos, surgindo assim novas inversões de marcha, 52% em relação às inversões iniciais. No total, a média do número de inversões de marcha efectuadas nas redes com 17 nodos passou de 3,1 para 1,8, ou seja, diminuiu, ainda assim, 42%. Relativamente aos valores mínimo e máximo houve uma redução de 81,8% e de 57,7%, respectivamente.

Em relação ao número total de nodos onde se efectuam inversões de marcha, o mesmo passou de 2,6 para 1,5, tendo-se então registado uma diminuição de 42%.

Salienta-se o facto de que o objectivo destes testes é apenas a eliminação das inversões de marcha efectuadas pelo algoritmo *Inspecção*, como se na realidade fossem inversões de marcha proibidas de efectuar. Observa-se que os resultados obtidos são bastante bons, dado que foi possível eliminar 93,5% das inversões de marcha iniciais.

Nas tabelas seguintes apresentam-se os resultados obtidos nos testes realizados para redes com 26, 37 e 50 nodos. As tabelas são análogas às anteriores, ou seja, cada

conjunto de oito testes realizado para o mesmo tipo de rede (com o mesmo número de nodos) encontra-se descrito em três tabelas.

Tabela 4.4 – Resultados do Algoritmo *Inspecção* em redes de 26 nodos.

Problema teste	N.º de arcos	N.º médio de viragens	N.º médio de		Total de más viragens (%)
			viragens à esquerda	inversões de marcha	
9	50, 54	51	2,4	3,6	11,8
10	55, 56	55	4,6	4,6	16,7
11	59, 60	58	3,6	4,6	14,1
12	62, 64	62	4,4	5,6	16,1
13	65, 66	64	4,4	5,8	15,9
14	67, 69	67	7	4,6	17,3
15	70, 73	70	5,6	7,6	18,9
16	74, 82	77	2,2	6,2	10,9
<i>Média</i>	—	—	4,3	5,3	15,2
<i>Mínimo</i>	50	51	2,2	3,6	10,9
<i>Máximo</i>	82	77	7	7,6	18,9

Nesta tabela verifica-se que o número de arcos para as redes de 26 nodos varia entre 50 e 82. Para estas redes, tem-se, novamente, os valores médios, mínimos e máximos, das inversões de marcha superiores aos das viragens à esquerda.

Em termos do número de viragens efectuadas nos CE's, em média, 15,2% das mesmas são consideradas más, valor este melhor do que o verificado para as redes de 17 nodos, tanto em valores médios, como em valores mínimos e máximos.

Tabela 4.5 – Viragens à esquerda em problemas com 26 nodos.

Problema teste	N.º de arcos	Viragens à esquerda		Variação das vir. esq. (%)
		Alg. Inspeção	Rede Expandida	
9	50, 54	2,4	1,6	-33,3
10	55, 56	4,6	5,2	+13
11	59, 60	3,6	2,8	-22,2
12	62, 64	4,4	5	+13,6
13	65, 66	4,4	4,2	-4,5
14	67, 69	7	5,4	-22,9
15	70, 73	5,6	4,4	-21,4
16	74, 82	2,2	1,8	-18,2
Média	—	4,3	3,8	-12
Mínimo	50	2,2	1,6	-33,3
Máximo	82	7	5,4	13,6

Da tabela 4.5 pode observar-se que o número de viragens à esquerda nem sempre diminuiu com a expansão da rede. Em termos médios verificou-se uma diminuição de 12% das viragens à esquerda. No entanto, há casos em que o número de viragens à esquerda aumenta, atingindo valores de 13% e 13,6.

Note-se ainda que o valor absoluto do valor mínimo, 33,3%, corresponde, de facto, à redução máxima do número de viragens à esquerda e o valor máximo, 13,6%, corresponde, por sua vez, ao aumento máximo do número de viragens à esquerda.

Tabela 4.6 – Inversões de marcha em problemas com 26 nodos.

Problema teste	Algoritmo <i>Inspecção</i>		N.º de nodos expandidos	Algoritmo <i>Altera CE</i>		
	N.º de inversões de marcha	N.º de nodos com inversões		N.º de inversões mantidas	novas	N.º de nodos com inversões
9	3,6	3	3	0	2,4	1,8
10	4,6	4,2	4	0,4	1,6	1,8
11	4,6	4,2	4	0,8	2	2,4
12	5,6	4,4	4,2	0	3,2	2,8
13	5,8	4,8	4,2	0,6	3	3,2
14	4,6	4,4	4,2	0	2,4	2
15	7,6	7,2	7	0,2	2	1,6
16	6,2	5	5	0,6	3	3,2
<i>Média</i>	5,3	4,7	4,5	0,3	2,5	2,4
<i>Mínimo</i>	3,6	3	3	0	1,6	1,6
<i>Máximo</i>	7,6	7,2	7	0,8	3,2	3,2

Nesta tabela observa-se que foram expandidos 96% dos nodos onde se realizavam inversões de marcha.

Mantiveram-se 5,7% das inversões de marcha efectuadas com o algoritmo *Inspecção* e surgiram novas inversões de marcha, 47,2% em relação às inversões iniciais. No total, o número de inversões de marcha efectuadas nas redes geradas com 26 nodos, em média, passou de 5,3 para 2,8, ou seja, diminuiu, ainda assim, 47,2%.

Em relação ao número total de nodos onde se efectuam inversões de marcha, o mesmo passou de 4,7 para 2,4, ou seja, diminuiu cerca de 49%.

Observa-se que os resultados obtidos continuam bastante bons, dado que se eliminou 94,3% das inversões de marcha.

Tabela 4.7 – Resultados do Algoritmo *Inspecção* em redes de 37 nodos.

Problema teste	N.º de arcos	N.º médio de viragens	N.º médio de		Total de más viragens (%)
			viragens à esquerda	inversões de marcha	
17	77, 81	78	7,6	6	17,4
18	81, 83	81	6,8	6,6	16,4
19	85, 87	85	7,2	8,2	18,1
20	88, 89	88	7,4	8,4	17,9
21	90, 91	90	8,6	8,6	19,2
22	92, 95	93	7,6	7,2	15,9
23	96, 100	97	5	7,8	13,2
24	101, 108	103	7,8	7,8	15,2
<i>Média</i>	—	—	7,3	7,6	16,7
<i>Mínimo</i>	77	78	5	6	13,2
<i>Máximo</i>	108	103	8,6	8,6	19,2

Para problemas com 37 nodos o número de arcos varia entre 77 e 108. Os valores médios e mínimos das inversões de marcha continuam superiores aos das viragens à esquerda mas os valores máximos são, para estes testes, iguais.

Em termos do número de viragens efectuadas nos CE's, em média, 16,7% das mesmas são consideradas más.



Tabela 4.8 – Viragens à esquerda em problemas com 37 nodos

Problema teste	N.º de arcos	Viragens à esquerda		Redução das vir. esq. (%)
		Alg. Inspeção	Rede Expandida	
17	77, 81	7,6	5,6	26,3
18	81, 83	6,8	5,4	20,6
19	85, 87	7,2	6,2	13,9
20	88, 89	7,4	4,4	40,5
21	90, 91	8,6	6,6	23,3
22	92, 95	7,6	5,8	23,7
23	96, 100	5	4,2	16
24	101, 108	7,8	6,4	17,9
Média	—	7,3	5,6	22,8
Mínimo	77	5	4,2	13,9
Máximo	108	8,6	6,6	40,5

Para as redes geradas com 37 nodos o número de viragens à esquerda voltou a diminuir sempre com a expansão da rede. Essa diminuição, 22,8%, pode considerar-se significativa.

Tabela 4.9 – Inversões de marcha em problemas com 37 nodos.

Problema teste	Algoritmo <i>Inspecção</i>		N.º de nodos expandidos	Algoritmo <i>Altera CE</i>		
	N.º de inversões de marcha	N.º de nodos com inversões		N.º de inversões mantidas	novas	N.º de nodos com inversões
17	6	5,2	4,8	0,2	3,2	2,8
18	6,6	5,4	5	0	1,6	1,4
19	8,2	7,4	7,2	0,4	2,6	2,4
20	8,4	7	7	0,8	3,4	3,8
21	8,6	7,2	6,6	0,6	3,8	3,4
22	7,2	6	5,8	0,8	2,4	2,4
23	7,8	6	6	0	2,2	2,2
24	7,8	7,2	7,2	0	3	2,6
<i>Média</i>	7,6	6,4	6,2	0,4	2,8	2,6
<i>Mínimo</i>	6	5,2	4,8	0	1,6	1,4
<i>Máximo</i>	8,6	7,4	7,2	0,8	3,8	3,8

Relativamente aos problemas teste em redes com 37 nodos, pode observar-se, da tabela 4.9, que foram expandidos 96,9% dos nodos onde se realizavam inversões de marcha.

Mantiveram-se 5,3% das inversões de marcha efectuadas com o algoritmo *Inspecção*, tendo surgido novas inversões de marcha, 36,8%, em relação às inversões iniciais. No total, o número de inversões de marcha efectuadas nas redes destas dimensões, em média, passou de 7,6 para 3,2, ou seja, diminuiu, 57,9%.

Em relação ao número total de nodos onde se efectuam inversões de marcha houve uma diminuição de 65,8%, pois este passou de 7,6 para 2,6. Note-se que estas reduções, em termos percentuais, são superiores às verificadas nos testes das redes anteriores.

Tabela 4.10 – Resultados do algoritmo *Inspecção* em redes de 50 nodos

Problema teste	N.º de arcos	N.º médio de viragens	N.º médio de		Total de más viragens (%)
			viragens à esquerda	inversões de marcha	
25	104,113	108	9,4	8	17,4
26	114,116	115	8	11	19
27	118,120	118	11	10,8	21,8
28	121,122	120	9,2	10,2	19,4
29	125,127	125	10,4	10,2	20,8
30	128,131	128	9,8	13	22,8
31	132,133	131	9,6	10,8	20,4
32	135,138	136	10,2	10,8	21
<i>Média</i>	—	—	9,7	10,6	20,3
<i>Mínimo</i>	104	108	8	8	17,4
<i>Máximo</i>	138	136	11	13	22,8

O número de arcos para as redes geradas com 50 nodos, varia entre 104 e 138. Os valores médios e máximos das inversões de marcha são superiores aos valores das viragens para a esquerda. Os valores mínimos para estes problemas teste são iguais.

Note-se que, em média, 20,3% das viragens efectuadas nos circuitos são consideradas más.

Tabela 4.11 – Viragens à esquerda em problemas com 50 nodos

Problema teste	N.º de arcos	Viragens à esquerda		Redução das vir. esq. (%)
		Alg. Inspeção	Rede Expandida	
25	104,113	9,4	7,6	19,1
26	114,116	8	6,2	22,5
27	118,120	11	9,4	14,6
28	121,122	9,2	7	23,9
29	125,127	10,4	9	14,3
30	128,131	9,8	9,2	6,1
31	132,133	9,6	8,8	8,3
32	135,138	10,2	9,4	7,3
Média	—	9,7	8,3	14,5
Mínimo	104	8	6,2	6,1
Máximo	138	11	9,4	23,9

Nesta tabela pode observar-se que o número de viragens efectuadas para a rua mais à esquerda, mais uma vez, desceu sempre com a expansão da rede e essa redução pode considerar-se significativa, cerca de 14,5%. Tanto o valor médio, como o mínimo e o máximo, são, para estas redes, inferiores aos verificados para as redes de 17 nodos e de 37 nodos.

Tabela 4.12 – Inversões de marcha em problemas com 50 nodos.

Problema teste	Algoritmo <i>Inspecção</i>		N.º de nodos expandidos	Algoritmo <i>Altera CE</i>		
	N.º de inversões de marcha	N.º de nodos com inversões		N.º de inversões mantidas	novas	N.º de nodos com inversões
25	8	7	7	0,2	2,6	2,4
26	11	10	9,2	0,2	4,2	4
27	10,8	9,8	9	0,8	3,2	3,6
28	10,2	9,2	9	1,8	2,2	3
29	10,2	8,6	8	0,8	5	4,2
30	13	11,2	11,2	0,4	6,2	6
31	10,8	9,4	8,8	1,2	3,4	4,4
32	10,8	10,6	10,2	1	5,6	5,8
<i>Média</i>	10,6	9,5	9,1	0,8	4	4,2
<i>Mínimo</i>	8	7	7	0,2	2,2	2,4
<i>Máximo</i>	13	11,2	11,2	1,8	6,2	6

Da tabela 4.12 pode observar-se que foram expandidos 95,8% dos nodos onde se realizavam inversões de marcha.

Após a expansão da rede, ainda se mantiveram 7,5% das inversões de marcha efectuadas.

No total, o número de inversões de marcha efectuadas nas redes geradas com 50 nodos, passou de 10,6 para 4,8, em média, ou seja, diminuiu 54,7%, tendo o número de nodos onde se efectuam essas inversões de marcha, diminuído cerca de 55,8%.

Nas duas tabelas seguintes resumem-se valores mais relevantes dos testes realizados para os quatro tipos de redes.

Assim, na tabela 4.13 apresentam-se os valores mínimos, médios e máximos da percentagem de más viragens efectuadas nos circuitos Eulerianos com o algoritmo *Inspecção* e com a rede expandida, ou seja, com os algoritmos *Expande Rede* e *Altera CE*, observados para cada conjunto de testes do mesmo tipo de rede.

Tabela 4.13 – Resumo do número de más viragens.

N.º de nodos	Percentagem de más viragens efectuadas em CE com					
	Algoritmo <i>Inspecção</i>			Rede expandida		
	mínima	média	máxima	mínima	média	máxima
17	12,4	16,4	25,7	3,9	10,3	16,6
26	10,9	15,2	18,9	7	10,5	13,2
37	13,2	16,7	19,2	6,6	9,8	12,2
50	17,4	20,3	22,8	9,2	10,7	12,3

Pode observar-se que com o algoritmo *Inspecção* (colunas 2, 3 e 4) a percentagem de más viragens, para os valores médios, variou entre 15,2 e 20,3%.

Observa-se também que a diferença entre os valores máximos e mínimos diminuiu à medida que o número de nodos da rede aumentou. Passou de 13,3, nas redes com 17 nodos para 5,4 nas redes com 50 nodos.

Depois da rede expandida, colunas 5 a 7, a percentagem de más viragens desceu significativamente, variando agora entre 9,8 e 10,7%. Apesar destes valores médios serem sensivelmente idênticos, verifica-se também que a diferença entre os valores máximos e mínimos desce à medida que aumenta o número de nodos das redes, passando de 12,7 para 3,1.

Os valores apresentados na coluna 6 são aqueles que mostram de facto a eficiência do método heurístico para o PMV, podendo-se mesmo afirmar que o valor de 10% para a percentagem média das viragens efectuadas num CE, é bastante razoável.

Analogamente, a tabela 4.14 resume os principais resultados relativos às inversões de marcha.

Tabela 4.14 – Resumo da alteração do número de inversões de marcha.

N.º de nodos	Alterações nas inversões de marcha com a expansão da rede								
	Percentagem de inversões de marcha proibidas			Percentagem de inversões de marcha totais			Percentagem da redução de nodos com inversões de marcha		
	min.	méd.	máx.	min.	méd.	máx.	min.	méd.	máx.
17	100	93,5	92,3	81,8	41,9	42,3	80	42,3	47,6
26	100	94,3	89,5	55,6	47,2	47,4	46,7	49	55,6
37	100	94,7	90,7	73,3	57,9	53,5	73,1	59,4	48,6
50	97,5	92,5	86,2	70	54,7	38,5	65,7	54,7	46,4

Assim, nas colunas 2 a 4 apresenta-se as reduções das inversões resultantes do algoritmo *Inspecção* verificadas por expandir a rede.

Estes valores mostram que nas redes de menores dimensões existem sempre casos em que se conseguem eliminar todas as inversões iniciais, sendo em média esta redução de 94%, aproximadamente. Nas redes com 50 nodos, maior dimensão testada, o número médio de inversões de marcha evitadas, 92,5% também se pode considerar elevado.

As colunas 5 a 7 da tabela 4.14 representam a percentagem de redução ao número total de inversões de marcha.

Este valor é calculado pelo quociente entre a diferença do número de inversões iniciais e finais e o número de inversões iniciais.

Por estes valores pode ver-se que as reduções nas inversões de marcha verificadas nas redes de 37 e 50 nodos, cerca de 56%, são superiores às verificadas nas redes de menores dimensões, 17 e 26, o valor verificado foi cerca de 45%.

As colunas 8 a 10 da tabela 4.14 representam a percentagem de redução ao número total de nodos onde se efectuam inversões de marcha.

Este valor é calculado pelo quociente entre a diferença do número de nodos com inversões iniciais e finais e o número de nodos com inversões iniciais.

Os valores da coluna 9 (redução dos valores médios do número de nodos com inversões de marcha), são muito semelhantes aos valores da coluna 6 (redução dos valores médios do número de inversões de marcha), pelo que se pode concluir que poucos são os casos em que se efectua mais do que uma inversão num nodo.

Capítulo 5. Conclusões

Este trabalho surgiu para aperfeiçoar e encontrar soluções, mais facilmente aplicáveis na prática, para um problema de optimização de rotas de recolha de resíduos sólidos urbanos.

O que se pretendeu nesta tese foi desenvolver um algoritmo que permita a obtenção de soluções em que se tenta evitar certo tipo de viragens, denominadas por más viragens.

Mesmo que as regras de trânsito não impeçam certas sequências de ruas na recolha, as inversões de marcha e as viragens à esquerda deverão ser evitadas, por serem mais perigosas.

Separando entre inversões indesejáveis e proibidas, foi apresentada uma modelização para o problema da recolha com proibições de inversão de marcha, utilizando-se para tal a expansão de uma rede Euleriana, definida a partir da rede de procuras.

Seguidamente, foi desenvolvido um algoritmo que permite encontrar um percurso com um baixo nível de viragens indesejáveis.

Para tal, foi identificado, numa rede Euleriana, um circuito Euleriano com um número reduzido de más viragens. A identificação deste circuito foi feita com uma alteração do algoritmo de Gondran e Minoux (1984) na escolha do próximo arco a considerar no circuito.

Para impossibilitar as inversões de marcha, é usada uma rede expandida nos nodos em que existiam as inversões proibidas.

O algoritmo *Identifica CE* foi testado em redes com 17 a 50 nodos e com 31 a 138 arcos, representativas de redes de estradas urbanas. Os resultados obtidos com os algoritmos, implementados em Pascal, permitiram concluir que o método é bom pois verificou-se uma diminuição significativa no número de más viragens. Como referido a expansão da rede permitiu reduzir, em média, para 10% a percentagem de más viragens efectuadas num circuito Euleriano final.

As inversões de marcha que se consideraram proibidas foram evitadas na sua maioria, situando-se entre os 92 a 95%, a percentagem de inversões de marcha proibidas que se conseguiram evitar.

As proibições às inversões de marcha impostas originam novas inversões no circuito, no entanto o total de inversões de marcha efectuadas no circuito, ainda assim, reduz cerca de 50%, em média.

Com este método verificou-se que nem sempre é necessário expandir os nodos onde existem inversões de marcha proibidas e que o número de viragens à esquerda também diminuiu, em geral, com a expansão da rede, pois em média diminuiu entre 12 a 30%.

Assim, o método aqui apresentado proporciona soluções que mais facilmente se aplicam na geração de rotas para os veículos que efectuam a recolha de resíduos.

As regras de trânsito impõem tanto proibições de inversão de marcha, como limitações a todo o tipo de manobras, como por exemplo, viragens à esquerda e à direita. Remete-se para investigação futura outros tipos de limitações de viragens.

Como trabalho futuro fica também em aberto a comparação dos resultados obtidos neste trabalho com os que se obteriam com outros métodos desenvolvidos para problemas semelhantes, como por exemplo, o método de McBride (1982).

Referências Bibliográficas

- Álvarez-Valdés, R.; Benavent, E.; Campos, V.; Corberán, A.; Mota, E.; Tamarit, J. e Valls, V. (1993), ARC. A Computerized System for Urban Garbage Collection, *Top*, 1, pp. 89-105.
- Ball, M.O.; Magnanti, T.L.; Monma, C.L. e Nemhauser, G.L. (1995), Network Routing, in: G.L. Nemhauser e A.H.G. Rinnooy Kan (ed.s), *Handbooks in Operations Research and Management Science*, Vol. 8, Elsevier Science B.V., North-Holland, Amsterdam.
- Beltrami, E.J. e Bodin, L.D. (1974), Networks and Vehicle Routing for Municipal Waste Collection, *Networks*, 4, pp. 65-94.
- Benavent, E.; Campos, V.; Corberán, A. e Mota, E. (1990), The Capacitated Arc Routing Problem. A Heuristic Algorithm, *Qüestió*, 14, pp. 107-122.
- Benavent, E.; Campos, V.; Corberán, A. e Mota, E. (1992), The Capacitated Arc Routing Problem: Lower Bounds, *Networks*, 22, pp. 669-690.
- Bodin, L.D. e Kursh, S.J.(1978), A Computer-Assisted System for the Routing and Scheduling of Street Sweepers, *Operations Research*, 26, pp. 525-537.
- Bodin, L.D. e Kursh, S.J.(1979), A Detailed Description of a Computer System for the Routing and Scheduling of Street Sweepers, *Computers and Operations Research*, 6, pp. 181-198.

- Bodin, L.; Fagin, G.; Welebny, R.; Greenberg, J. (1989), The Design of a Computerized Sanitation Vehicle Routing and Scheduling System for the Town of Oyster Bay, New York, *Computers and Operations Research*, 16, pp. 45-54.
- Caruso, C.; Colorni, A. e Paruccini, M. (1993), The Regional Urban Solid Waste Management System: A Modelling Approach, *European Journal of Operational Research*, 70, pp. 16-30.
- Chapleau, L.; Ferland, J.A.; Lapalme, G. e Rousseau, J.M. (1984), A Parallel Insert Method for the Capacitated Arc Routing Problem, *Operations Research Letters*, 3, pp. 95-99.
- Clark, R.M. e Gillean, J.I. (1975), Analyses of Solid Waste Management Operations in Cleveland, Ohio: A Case Study, *Interfaces*, 6, pp. 32-42.
- Eglese, R.W. e Li, L.Y.O. (1992), Efficient Routeing for Winter Gritting, *Journal of Operational Research Society*, 43, pp. 1031-1034.
- Eglese, R.W. e Li, L.Y.O. (1996), LOCSAR: A Local Search Algorithm for Constrained Arc Routing Problems, Presented at *IFORS96*, Vancouver, Canada.
- Ford, L.R. e Fulkerson, D.F. (1974), *Flows in Networks*, Princeton University Press, Sixth Printing.
- Golden, B.L. e Wong, R.T. (1981), Capacitated Arc Routing Problems, *Networks*, 11, pp. 305-315.
- Golden, B.L.; DeArmon, J.S. e Baker, E.K. (1981), Algorithms for the Capacitated Chinese Postman Problem, *Working Paper MS/S #81-024*, College of Business and Management, University of Maryland at College Park.

- Golden, B.L.; DeArmon, J.S. e Baker, E.K. (1983), Computational Experiments with Algorithms for a Class of Routing Problems, *Computers and Operations Research*, 10, pp. 47-59.
- Gondran, M. e Minoux, M. (1984), *Graphs and Algorithms*, John Wiley & Sons.
- Greistorfer, P. (1994a), Computational Experiments with Heuristics for a Capacitated Arc Routing Problem, *Working Paper*, N. 32, Department of Business, University of Graz, Austria.
- Li, L.Y.O. e Eglese, R.W. (1994), A Comparison of Simulated Annealing and Tabu Search for a Constrained Arc Routing Problem, Unpublished manuscript.
- Male, J.W. e Liebman, J.C. (1978), Districting and Routing for Solid Waste Collection, *Journal of Environmental Engineering Division*, 104 (EE1), pp. 1-14.
- Marks, D.H. e Liebman, J.C. (1970), Mathematical Analysis of Solid Waste Collection, *U.S. Department of Health, Education and Welfare, Public Health Service*, 2104, Bureau of Solid Waste Management.
- McBride, R. (1982), Controlling Left and U-turns in the Routing of Refuse Collection Vehicles, *Computers and Operations Research*, 9, pp. 145-152.
- Minieka, E. (1979), The Chinese Postman Problem for Mixed Networks, *Management Science*, 25, pp. 643-648.
- Mourão, M.C. (1997), *Optimização de Rotas na Recolha de Resíduos Urbanos. Modelos e Algoritmos*, Tese de doutoramento, MAEG, ISEG-UTL.

- Or, I. (1990), Improving the Efficiency of the Solid Waste Collection and Transportation System in IZMIR and Network Techniques, *Presented at IFORS 90*, Athens, Greece.
- Pearn, W.L. (1991), Augment-Insert Algorithms for the Capacitated Arc Routing Problem, *Computers and Operations Research*, 18, pp. 189-198.
- Roy, S. e Rousseau, J.M. (1988), The Capacitated Canadian Postman Problem, *Centre de Recherche sur les Transports*, #587, Université de Montréal, Canada.
- Saruwatari, Y.; Hirabayashi, R. e Nishida, N. (1991), Subtour Elimination Algorithm for the Capacitated Arc Routing Problem, *Technical Report*, Science University of Tokyo, Tokyo 162, Japan.
- Shuster, K.A e Schur, D.A. (1974), Heuristic Routing for Solid Waste Collection Vehicles, *US Environmental Protection Agency*, SW-113; U.S. Government Printing Office, Washington, D.C..
- Stern, H.I. e Dror, M. (1979), Routing Electric Meter Readers, *Computers and Operations Research*, 6, pp. 209-223.
- Turner, W.C. e Hougland, E.S. (1975), The Optimal Routing of Solid Waste Collection Vehicles, *AIIE Transactions*, 7, pp. 427-431

